Project No. 10-964

Methods and Tools to Allow Molecular Flow Simulations to be Coupled to Higher Level Continuum Descriptions of Flows in Porous/Fractured Media and Aerosol/Dust Dynamics

Fuel Cycle

Dr. Sudarshan K. Loyalka
University of Missouri-Columbia

Dan Funk, Federal POC
Keith Bradley, Technical POC

NEUP Nuclear Energy University Programs
U.S. Department of Energy

Final Report, NEUP-964
April 7, 2015

Methods and Tools to allow molecular flow simulations to be coupled to higher level continuum descriptions of flows in porous/fractured media and aerosol/dust dynamics

Principal Investigator: Sudarshan K. Loyalka
Institution: University of Missouri-Columbia
Collaborators: None

Duration: 4 years (10/1/2010-9/30-2014)
Total Funding Level: $541,287

Summary

The purpose of this project was to develop methods and tools that will aid in safety evaluation of nuclear fuels and licensing of nuclear reactors relating to accidents. The objectives were to develop more detailed and faster computations of fission product transport and aerosol evolution as they generally relate to nuclear fuel and/or nuclear reactor accidents. The two tasks in the project related to molecular transport in nuclear fuel and aerosol transport in reactor vessel and containment. For both the tasks, explorations of coupling of Direct Simulation Monte Carlo with Navier-Stokes solvers or the Sectional method were not successful. However, Mesh free methods for the Direct Simulation Monte Carlo method were successfully explored. These explorations permit applications to porous and fractured media, and arbitrary geometries. The computations were carried out in Mathematica and are fully parallelized. The project has resulted in new computational tools (algorithms and programs) that will improve the fidelity of computations to actual physics, chemistry and transport of fission products in the nuclear fuel and aerosol in reactor primary and secondary containments.

Table of Contents:

I. Introduction

The purpose of this project was to develop methods and tools that will aid in safety evaluation of nuclear fuels and licensing of nuclear reactors relating to accidents. The objectives were to develop more detailed and faster computations of fission product transport and aerosol evolution as they generally relate to nuclear fuel and/or nuclear reactor accidents. The two tasks in the project related to molecular transport in nuclear fuel and aerosol transport in reactor vessel and containment. The accomplishments of the project are as follows:

II. Tasks and Accomplishments

Task 1: Methods and tools to allow molecular flow simulations to be coupled to higher level continuum descriptions of flows in porous/fractured media

We focused here on molecular simulations (the non-linear Boltzmann equation for gas mixtures), continuum solutions (Navier-Stokes or higher order Burnett equations), and good understanding and use of asymptotics and domain decomposition methods for simultaneous coupling of the two. Central to our effort is the ability for molecular level computation of flows based upon the non-linear Boltzmann equation. Deterministic solutions of this equation are difficult, but insights have been gained through solutions of the linearized Boltzmann equation or the linear neutron transport equation, as well as model equations. Just like the Monte Carlo techniques in neutron transport, the DMSC technique has succeeded remarkably well. The molecular flows differ in two essential respects from neutron transport (where neutron-neutron interactions are not an issue) in that intermolecular collisions are a dominant feature (non-linearity), together with molecule-surface interaction (quasi-linear or linear). In DSMC, a population of molecules is evolved in time with respect to each molecule’s position, velocity, and state (orientation, internal energy etc. if needed), accounting for both intermolecular collisions and molecule surface interactions. Generally molecules are partitioned in spatial cells over the flow geometry. In each cell, all molecules are advanced (evolved) over a time step first for a free-molecular trajectory, and then, at the end, a number are sampled for collisions according to different Monte Carlo algorithms (herein lies the essence of the method with respect to speed). Physical properties such as density, velocity, temperature, stress tensor, etc. are the moments of the molecular distribution, and are computed for any desired cell from simple summations (sometime a distinction is made between computational and sampling cells to keep fluctuations under control).

Sample DSMC programs are available , but we focused on constructing and expanding upon on our own programs in *Mathematica* to enable greater flexibility, visualization, and range of exploration. We have constructed the *Mathematica* program, and used lists and list operations extensively to simulate molecular transport. The program is parallelized, and we have run it on several computers from one, two, four, six, eight to 128 processors. Running the program on a computer with arbitrary number of processors is straightforward and it would not require any further modifications. We addressed the issue of arbitrary geometries by using special routines in Mathematica8.0 (and onwards) on clustering of molecules in the list of molecules, we can make the computations mesh free (the clusters are used in place of cells, and are not fixed thus providing great flexibility in handling regions with small or large variations in properties). We have tested these computations against fixed mesh (cell) calculations for the parallel plate geometry, and the results between the two have been in agreement. We have run the program extensively for the classical problem of heat transfer between parallel plates for several choices of parameters.

We have made improved the program with respect to speed/storage. The speed of the program, which is strongly influenced by its memory overhead, is effectively limited by the total number of particles which can be simulated at one time. This speed limitation determines the overall precision of the results. Code modifications have been made which improve the overall speed of the program by 2 for an equal number of particles. This has been accomplished by implementing a number of changes that do not change the core logic of the program, but which take better advantage of some of Mathematica's functionality. Certain highly repetitive routines, specifically the particle movement algorithm, have been rewritten to take advantage of Mathematica's virtual machine by compiling those functions. This reduces the computational overhead for particle motion to data and list manipulation, not physics calculations. This has been where the majority of the speed increase has occurred and extending it to the collisional routine is an area of interest. The other issue we addressed was in the collisional step, which is the effective operating algorithm for continuum and near continuum flow regimes. In these flow regime each simulated particle represents significantly more particles than in the other flow regimes, which can necessitate more simulated particles in order to obtain useful results. The particle selection and collision routines were rewritten to convert the particle interaction process from an iterative process to a single pass operation for a cell during an individual time step. Several modifications have been made to improve the stability of our implementation of Bird's technique by exercising better control over the collision rate, which is governed by the maximum relative particle velocity in each cell. Averaging the number of particles in a region over several prior time steps stabilizes the number of particles selected for collision between time steps, decreasing the fluctuation of the results. Finally we made changes that consider the fractional nature of the number of particles selected for collision, so that this process does not arbitrarily introduce systematic error.

We were not successful in coupling the DSMC calculations to a Navier-Stokes solver. We however made very substantial progress in exploration of mesh-free calculations for DSMC. We have attached draft of a manuscript that describes our efforts. We will be submitting this manuscript with some revisions for publication in the summer 2015.

Task 2: Methods and tools to couple to DSMC and Sectional Techniques for Aerosol Dynamics

The purpose here was to develop methods for improving upon the simulation of aerosol dynamics by means of coupling the Direct Simulation Monte Carlo (DSMC) method and the sectional method. The MAEROS program makes use of a sectional technique to monitor the amount of mass in kilograms, , distributed amongst each component, *k*, within each diameter (or volume) section, ℓ. This program is used for CFD codes where aerosol loadings are large, and coagulation is significant. While this program has a great advantage in its computational speed and usefulness, it has been shownto be insufficient in many cases of multi-component aerosols due to its strong averaging over component densities.

In contrast, the DSMC program uses a sample of particles of various masses and component distributions from initial distribution. This population of particles is evolved in time accounting for various rate processes. Generally, a time step is chosen, and a number are sampled for collisions according to different algorithms. While DSMC maintains fidelity to actual physics for interactions, its computation time is often orders of magnitude greater than that of the sectional method.

The project objectives were then first to improve upon each of these programs by increasing their individual computation times and then also to couple them together in order to take advantage of the computational speed of MAEROS while still maintaining fidelity to the particle physics that DSMC provides. This coupled program should be dynamic in nature in that it should identify when MAEROS is insufficient for modeling the multi-component aspect of the aerosols and switch to DSMC for a more precise handling of the interactions. Also, each program makes use of very distinct sets of input so conversion programs are needed to take a mass distribution (as used by the sectional method) and construct a sample list of particles (DSMC) and vice versa. Care must be taken to ensure that in this conversion, as little information concerning mass and component distribution is lost as possible.

We succeeded in making both the Sectional (MARROS) and DSMC computations (we carried out all computations in Mathematica) efficient. This was accomplished by, first, improving upon the implementation of the sectional technique in particular. Our improved version of MAEROS is capable of running a simulation in 20 seconds as compared to the hour long computation time at the beginning of the project. Second, the needed conversion programs were created. The conversion programs are capable of going either direction in converting between MAEROS and DSMC data formats with very little loss in mass. We were however not able to couple the two techniques well. Our efforts are fully described in our attached paper that has been accepted for publication in Nuclear Science and Engineering.

In addition to the above, guided by our efforts in DSMC computations of molecular flows, we also explored mesh-free computations of aerosol dynamics. We have been successful here, and we have prepared a manuscript (draft, attached) that we will submit for publication in spring, 2015.

DRAFT of a Manuscript

Explorations of Mesh Free DSMC Computations for Molecular Flows

John F. Palsmeier, Shawn Campbell and Sudarshan K. Loyalka*
Nuclear Science & Engineering Institute,
Particulate Systems Research Center
University of Missouri
Columbia MO 65211

* Corresponding Author
Complete Mailing Address:
Nuclear Science & Engineering Institute
Particulate Systems Research Center
E 2433 Lafferre Hall
University of Missouri
Columbia, MO 65211 USA
email: LoyalkaS@missouri.edu
phone: (573) 882-3568
fax: (573) 884-4801

Number of pages: 37
Number of tables: 5
Number of figures: 27

Abstract

The behavior of gas/vapor flows in porous mediums such as nuclear fuels is a topic of interest as the current methods available leave room for improvement. The current continuum methods do not suffice as the time and length scales under consideration vary over a large range.

The Direct Simulation Monte Carlo (DSMC) technique is an effective tool for simulating molecular gas dynamics. We have used this technique to investigate mass and energy transfer under flow conditions of relevance to nuclear fuels by implementing mesh free techniques to overcome some of the limitations of cell based methods.

KEYWORDS: Mesh-Free DSMC

I. INTRODUCTION

Gas and vapor flows occur in porous and fractured media which can be difficult to accurately describe using current methods. This issue is of particular concern with regard to the behavior of nuclear fuels over the course of the fuel cycle as they undergo various forms of degradation due to thermal, chemical, and radiological causes. The use of continuum based descriptions for fission product transport in nuclear fuels is insufficient when the large range of the time and length scales under consideration is taken into account. As such other methods must be used to supplement or replace the continuum descriptions for fission product transport.

The implementation of molecular simulation, which is governed by the non-linear Boltzmann equation for gas mixtures, is presented in this work as a method for solving the aforementioned problem. Deterministic solutions are inadequate for all but the simplest problems and while numerical methods can be used for more complicated cases, they generally require the introduction of assumptions for the sake of obtaining a solution. One technique which has been successful in similar problems is Direct Simulation Monte Carlo (DSMC), which has been used for describing the behavior of fluid flows in a number of different applications, including molecular and aerosol flows in nuclear systems. DSMC is similar in to the Monte Carlo methods used in simulating neutron transport, with two key differences. These are that intermolecular collisions are a dominant feature (non-linearity) while neutron-neutron collisions are generally neglected, and that molecule-surface interactions are a significant feature while in neutron transport a change in surface generally only signifies a change in the transport medium properties.

The standard method for defining the geometrical properties of the volume under consideration in order to implement the appropriate boundary conditions for systems which cannot be adequately solve analytically is to mesh the volume. Meshing is the discretization of a volume into smaller volumes for the purpose of solving a system of equations by localizing interactions or simplifying the equations. In the case of DSMC it is used to ensure that particles are only interacting with their neighbors. When solving the Navier-Stokes equation using CFD techniques, meshing is used so that the governing differential equations can be converted into a system of algebraic equations.

We have implemented a mesh-free simulation using the DSMC technique. The ability to run a simulation without having to compute cell geometries either initially or continuously has the potential to greatly improve runtime and to free the simulation results from the quality of the initial mesh. By mesh quality we mean the degree to which the mesh used is able to adequately represent the spatial variability in a system, particularity in regions with large time or spatial gradients relative to the rest of the system.

Our purpose in this paper is to report on our exploration of this mesh-free DSMC implementation for simulating mass and energy transfer of molecular gasses of relevance to nuclear fuels as a means of overcoming the limitations of mesh/cell based methods. In the second we discuss our implementation of the mesh-free method. In the third section we describe some example problems and present our results from running simulations.

II. MESH FREE METHOD

The benefits of the DSMC technique over other solution methods and its implementation are discussed in detail in a number of sources. The focus here is on the significance of the mesh-free method we have developed over the traditional cell based method used in both the DSMC technique and in numerical solution methods. The basic limitation of mesh based methods is that their effective quality is limited by their generating function and the time involved in computing the mesh. In complex systems the mesh must undergo an iterative refinement process in order to determine the regions where large gradients exists that need a finer mesh to achieve the desired level of accuracy. When the system has time dependant component involved the initially refined mesh may no longer be adequate and a new mesh must be generated. This leads to a considerable quantity of computational resources being directed at continuously re-meshing the system, or a computation with results whose accuracy can be suspect.

The method we have developed for a mesh-free simulation must still contain a volumetric component in order to calculate particle number density based behaviors such a molecule-molecule collisions. This is accomplished by dynamically creating volume groups utilizing a type of nearest neighbor technique. This clusters particles based on selected properties, which in this case are their spatial coordinates. In addition the nearest neighbor method insures that particles are not excluded from interacting with each other by arbitrary geometric boundaries created from meshing.

Mathematica ® version 9 was the primary software employed in this research as we have found its dynamic list handling features to be beneficial in developing DSMC programs. The function used to create the particle groups or clusters is a built in Mathematica function called

FindClusters. We will summarize the methodology by which it operates and then proceed to our technique for implementing the information obtained through clustering.

The cluster analysis performed by the software is accomplished by utilizing a variety of techniques to categorize data according to an unsupervised learning technique.[1] This permits the classification of the data without having pre-existing knowledge of its structure. The particular method we chose to implement is called "Optimize" in Mathematica, it is generally known as partitioning around medoids, which itself is a subset of the k-medoids algorithm. A number *k* of individual data points are selected at random as the starting points for each cluster. A distance metric is used to determine the effective dissimilarity of different data points, with an iterative process undertaken to minimize the dissimilarity function. In our application of this method the distance metric used is Euclidean distance, though other metrics exist. This implementation for a mesh-free simulation is exclusive to a Monte Carlo approach as it requires the spatial coordinates of a particle to compute the dissimilarity function, as opposed to the equation describing a distribution used in sectional methods.

The number of dimensions over which the clustering of data points should be carried out can be determined by considering the geometry and fluid dynamics of the system under consideration. In general, for any system of sufficient complexity that would benefit from the clustering method the distance metric should be calculated over all spatial dimensions. However, in the third section we include some examples of clustering over a single dimension for the purpose of illustrating where this approach is applicable. Such as in the case of heat transfer between two infinite parallel plates, where clustering over the dimension of heat transfer can be used to reduce the total number of clusters involved in the simulation.

Aside from the calculations involved in determining which cluster a particle belongs to is the issue of determining, for a given data group, the total number (k) of clusters. The appropriate value of k to choose can be difficult to determine without pre-existing knowledge of the data set distribution. This is because in general increasing the value of k will decrease the error associated with clustering the data set. However this generally leads to a decline in computational efficiency, so keeping this value low relative to the total number of molecules is crucial to minimizing simulation run time. While a number of methods exist for calculating an appropriate value of k, the majority of them require clustering over a range of k values by an iterative process. These methods are useful when data sets exhibit a degree of similarity so that a value of k can be determined a single time. However, in the dynamic systems we are considering this is computationally inefficient and we instead implement a rule of thumb which can rapidly changed based on the number of molecules in the system. This is given by:

$k \approx (n/2)^{1/2}$,

where n is the number of molecules in the simulation.

After determining the clustering method and the number of clusters, the final modification to the standard DSMC technique can be implemented. In general the particle density of a group of particles must be known in order to calculate particle interaction. Since the number of particles in a cluster and their positions are known, the effective particle density for a molecule cluster can then be found after determining the volume of the cluster. This is accomplished by determining which molecules and corresponding surfaces make up the convex hull of the cluster. The computation of the convex hull is a basic problem in computational geometry and detailed descriptions of this method can be found in (REF). In our computations we utilized Mathematica’s built-in convex hull algorithm. A sample figure of a convex hull and all the corresponding enclosed points is shown in Figure 1.

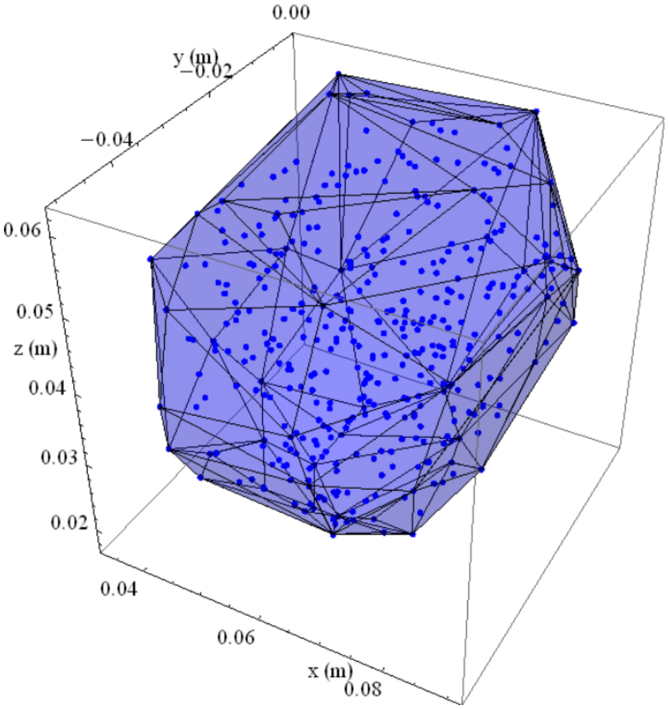


Figure 1. Convex hull of a cluster with molecules located at vertices and distributed through the volume.

The total volume of an individual cluster is then computed by considering the volume enclosed by each surface triangle if the triangle described three points of a tetrahedron, with the fourth point at the origin of the coordinate system for convenience. The equation for the volume of a tetrahedron is given by

$$V = \frac{\left|(a-d)\bullet((b-d)\times(c-d))\right|}{6},$$

where *a*, *b*, *c*, and *d* are vector coordinates. The fourth vector coordinate *d* is set to always correspond to the origin, thereby reducing the equation to

$$V = \frac{\left|a\bullet(b\times c)\right|}{6}.$$

However, by considering this equation with the absolute value operator removed a “negative” volume becomes possible. This negative volume corresponds with convex hull triangles whose outer surface faces the origin. When summing the volume contributions of all the surface triangles the volumes of the tetrahedrons whose outer surface triangles face away from the origin enclose the cluster, along with the volume between the cluster and the origin. The negative volume tetrahedrons then remove this extra volume, leaving only the cluster volume. The final equation for the volume of a cluster with *N* surfaces is then given by

$$V = \sum_{i=1}^{N} \frac{a_i \bullet (b_i \times c_i)}{6}.$$

Figure 2 illustrates the negative and positive volume contributions, where one of the triangle surfaces of the red cluster convex hull is projected to the origin. This forms the green tetrahedron, which encloses both part of the cluster volume and a volume outside the cluster that would be subtracted by a full volume calculation of the cluster.

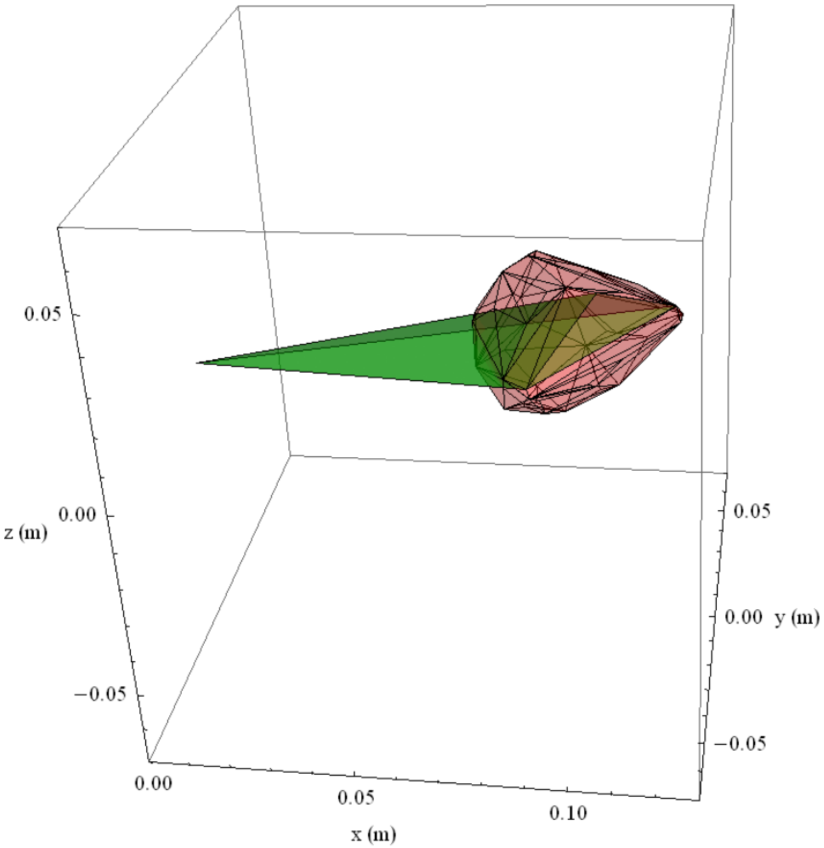


Figure 2. Tetrahedron formed by a cluster surface triangle and the origin.

III. TYPICAL PROBLEMS, RESULTS AND ANALYSIS

For the purposes of testing the simulation code developed, several typical problems are considered to include a variety of geometries and flow conditions. Specifically the heat transfer between two parallel plates and the matter transfer of gas flowing down a helical tube.

a. Parallel Plates

In cases where the direction of energy flow is one-dimensional the clustering technique leads to more accurate results when the clustering parameter is restricted to the direction of energy flow. This can be seen in Figure 3. In both cluster cases and in the mesh/cell case the same number of cluster groups/cells was utilized for the sake of an appropriate comparison. Increasing the number of clusters should bring the multidimensional cluster parameter group in line with the other results, as we believe it is suffering from poor resolution in this particular case. This would improve the generality of the simulation, but in this particular case clustering over multiple dimensions is computationally inefficient.

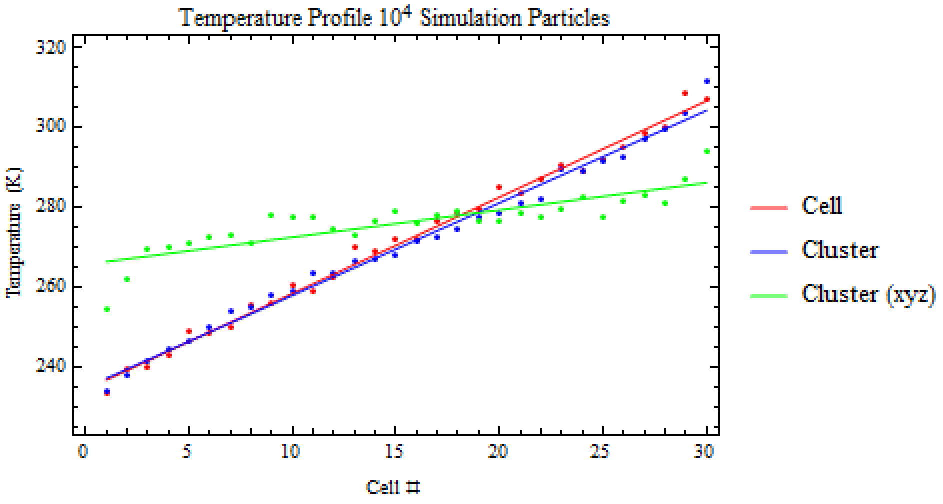


Figure 3. Temperature profile of Kn=0.1 parallel plate system using different techniques.

After implementing a significant number of changes we would like to present the results obtained and compare them with earlier results. In Table 1 we show that the changes made to improve computational efficiency and implement clustering have not substantially changed our results except in the KN = 1.0 case, where our results now approach the moments method results.

Table 1. Comparison of results by method and programming changes.

Distance between the plates (1/KN)	Heat Flux, W/m^2 (Moments Method)	Heat Flux, W/m^2 (10^4 particles * 48 Runs, Cell Method)	Heat Flux, W/m^2 (10^4 particles * 48 Runs, Cluster "X" Method)
0.1	34.40	32.99±0.28	33.02±0.29
1.0	29.94	26.21±0.29	29.62±0.30
10.0	12.98	9.27±0.28	9.25±0.29

The temperature distribution for these three cases is shown in Figure 5. The mesh-free technique was used in this case, the cell number only refer to the temperature profile across the gap. They are comparable to our past results as shown in Figure 6 for both the Kn=0.1 and 10 regimes, with some differences between the Kn=1 regimes. This is to be expected based on the change in the heat flux results.

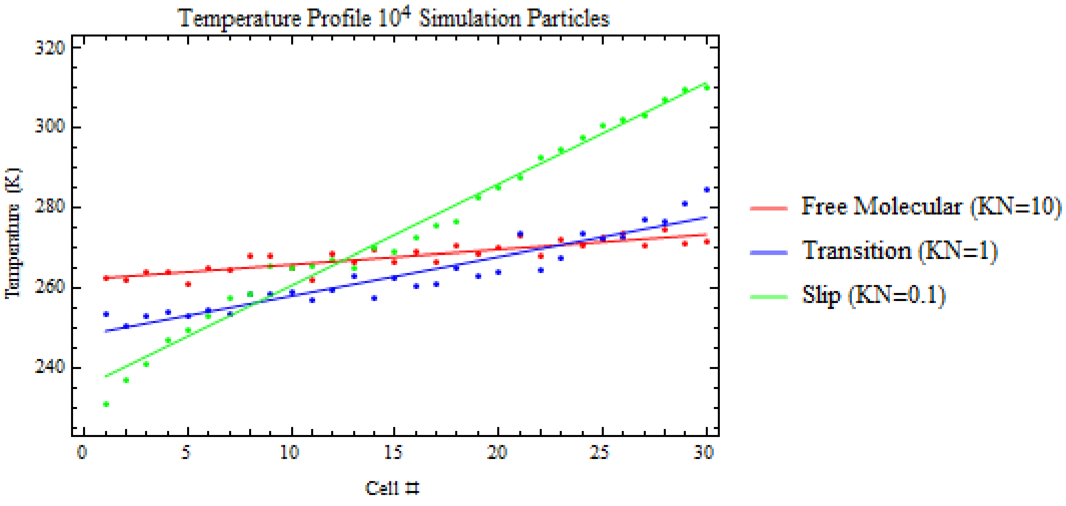


Figure 4. Temperature profile for three different flow regimes using mesh-free algorithm and speed modifications.

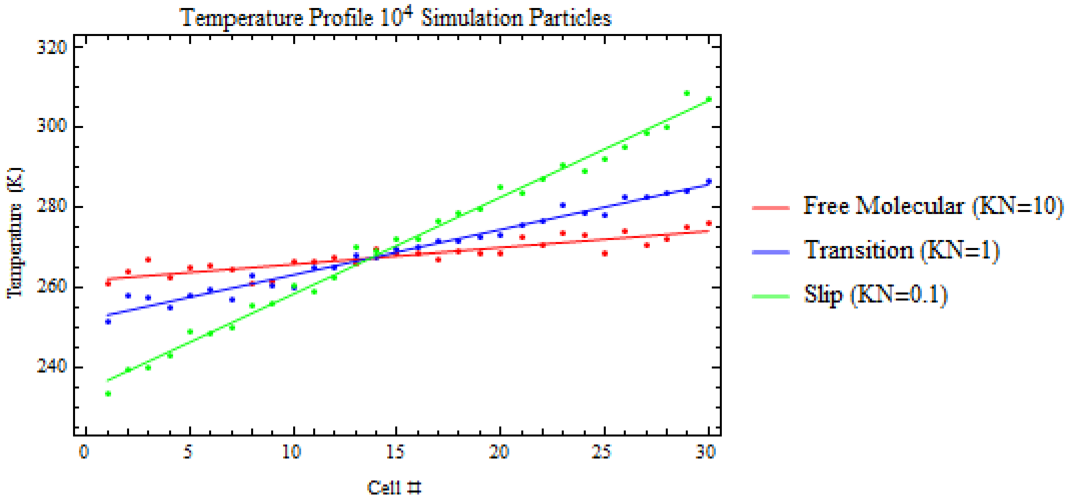


Figure 5. Temperature profile for three different flow regimes using previous methods.

b. Arbitrary Geometry

More complex geometries such as helical tubes are of interest in our studies as curved pathways are a common occurrence in fission product transport through spent fuel. We also chose a helical tube due to the greater degree of difficulty involved in implementation. A simple geometry such a cylinder or sphere is relatively easy to implement into the existing code, so while the results would not be trivial, further effort is better directed toward a complex geometry now that the underlying algorithm has been benchmarked. An example helical tube is shown below in Figure 7.

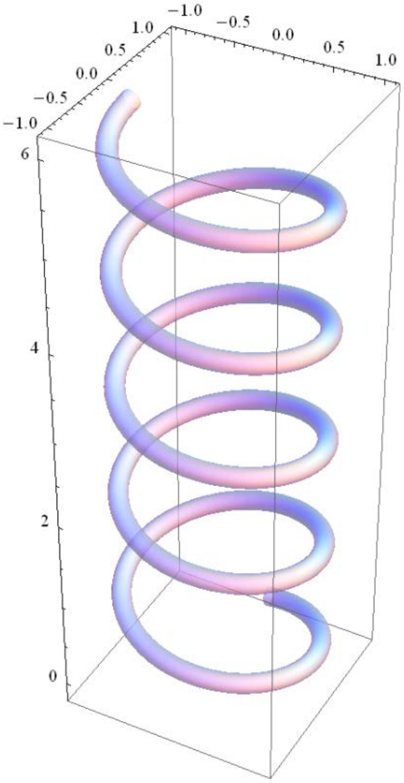


Figure 6. Example helical tube.

For the purpose of development and testing we used a short segment of this tube so we could reduce the initial computational overhead involved in the debugging process and to examine more closely the flow properties at the ends of the tube. The segment used for this process is shown in Figure 8, it has a path radius of 1 (m) corresponding to a path length of π (m), and a tube radius of 0.1 (m).

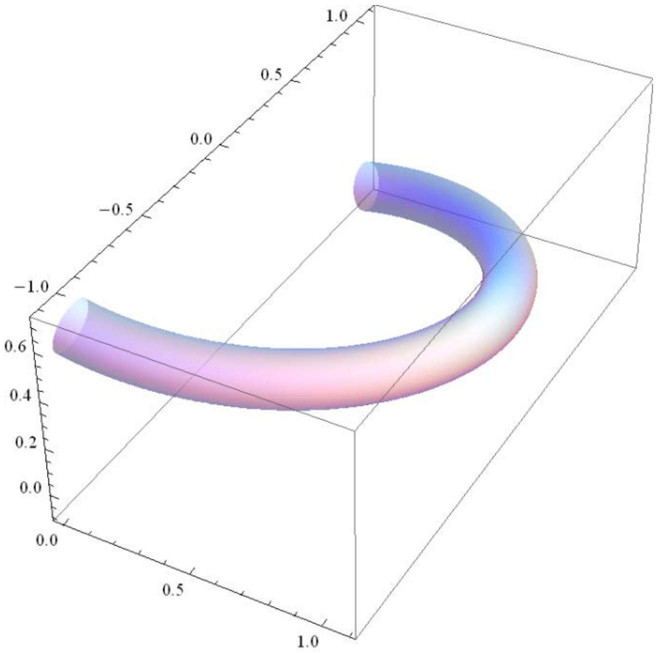


Figure 7. Helical tube used in developing wall interaction code.

The primary issue with implementing this geometry is in determining the intersection of a molecule with the wall of the tube. The helical tube must be, and the molecule trajectory can be, described by parametric equations. The Cartesian coordinate parametric equation for the helix is given by:

$$r(t) = (A\ Sin(t), ACos(t), B\ t).$$

Where r is the vector coordinate of the helix, t is the parameterization variable, A is the radius of the helix about its path curvature, and B is the pitch or height change per rotation. To describe the tube itself we have:

$$h(t,\theta) = r(t) + R(-Cos(\theta)\hat{n} + Sin(\theta)\hat{b}).$$

Where h is a point on the surface of the tube, R is the constant radius of the tube, θ is the angle of rotation about the helix centerline from 0 to 2 π, and $\hat{\boldsymbol{n}}$ and $\hat{\boldsymbol{b}}$ are the normal and binormal vectors of the helix centerline for that value of t as shown in Figure 9.

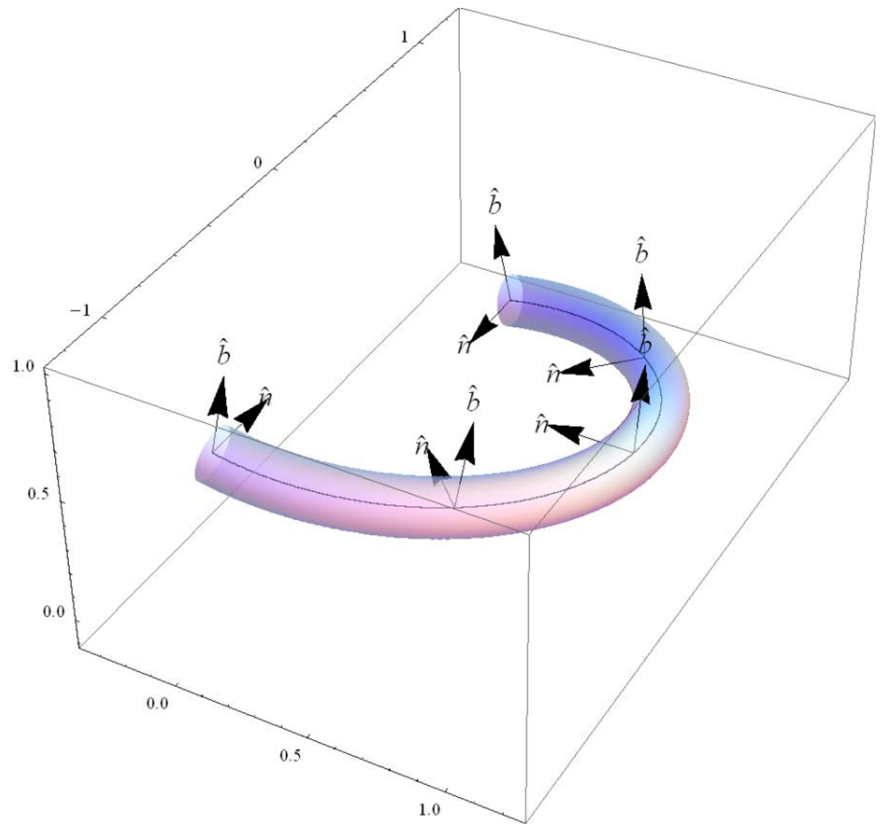


Figure 8. Position dependence of normal and binormal vectors.

The equation for the molecule's trajectory is given by:

$$l(d) = (v_x\ d + x, v_y\ d + y, v_z\ d + z)$$

Where *l* is the vector coordinate of a new position along the molecule trajectory, each *v* term is a component of the velocity unit vector, *d* is the product of the velocity scalar and the time step, resulting in a distance traveled, and *x*, *y*, and *z* are the previous coordinates.

The equations describing the tube and the molecule trajectory do not permit an analytical solution, so the coordinates must be found numerically, which introduces complications in the implementation of the required algorithms. The numerical method employed uses an iterative root finding technique. Three equations with three unknowns are then created by setting the *x*, *y*, and *z* components of *l*(*d*) *and h*(*t*,*θ*) equal to each other and solving for *d*, *t*, and *θ*. However we have found a more efficient system of equations to solve. Instead of solving for a line equation and a surface equation it is possible to obtain the desired coordinates by solving a different set of equations obtained by using the two line equations *l*(*d*) and *r*(*t*). These equations are converted to a distance function between a point on the trajectory path and a point on the helix centerline, with the distance between them set at radius *R*, the second equation minimizes the first equation by taking the first derivative with respect to *t* and setting it to zero as shown below

$$F(d,t) = (v_x\ d + x - A\ Sin(t))^2 + (\ v_y\ d + y - ACos(t))^2 + (v_z\ d + z. - B\ t)^2 - R^2\ = 0$$

and

$$\frac{dF(d,t)}{dt} = -2A\ Cos(t)\,(v_x\ d + x - A\ Sin(t)) + 2A\ Sin(t)\,(\ v_y\ d + y - ACos(t)) - 2B\ (v_z\ d + z. - B\ t) = 0$$

This results in restricting the equations to describing two points *R* meters apart with one end on the helix surface at the point of intersection, and the other point on the helix centerline at the point of closest approach. These equations are solved for *d* and *t*, which can then be used to find the coordinates of the intersection and to compute a coordinate system with respect to the local surface for velocity calculations.

The root finding method only gives the first root found, because multiple path intersections can occur it is important to use the appropriate search parameters. Initially we considered determining the point at which the intersection would occur and then determining if this would happen during the current time step; however, this led to errors and unstable results. We now determine whether the particle will interact with a surface based on its current unimpeded velocity, then determine the point of intersection.

Now that the point of intersection is known, the new velocity of the particle as it diffuses off of the surface can be calculated. The normal with respect to the surface can be found by determining the unit vector which starts at the point of intersecting and is directed at the point on the helix centerline of closest approach, which corresponds with the already determined value of parameter *t*. A tangent vector is found by recognizing that the tangent of the centerline at *r*(*t*) is also the tangent of any point on the surface whose normal is directed at *r*(*t*). The bitangent is then calculated by taking the cross product of the previous two vectors. Using this localized coordinate system a new velocity for the particle is selected according to the Maxwell-Boltzmann distribution. The local surface unit vectors for several surface points are shown in Figure 10.

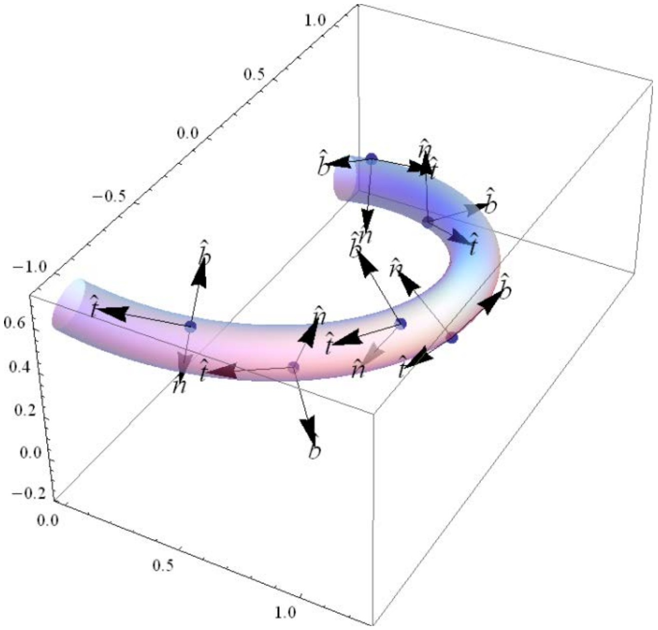


Figure 9. Normal, tangent, and bitangent vectors for several difference wall positions.

Unlike our previous system which was closed, this is an open system; therefore we have implemented tracking of particles that have flowed out the ends of the tube. In addition we are currently implementing a source term which will allow us to evaluate this system with a variety of conditions for both transient and equilibrium flows. We are currently considering a diffuse flow originating at one end of the tube which is then permitted to propagate through the volume. We are assuming a constant pressure at one end of the tube of 0.5 Pascals. In order to maintain this pressure a certain level of flux into the end of the tube must be maintained. In cases where the stream velocity u is zero or very small compared to the average molecular velocity $\bar{v}$ the flux into the tube is given by:

$$J^{+} = \frac{n\bar{v}}{4}, \text{ where } n = \frac{p}{k\,T}, \text{ and } \bar{v} = \sqrt{\frac{8k\,T}{\pi\,m}}.$$

J^{+} (#/cm^2 s) is the flux tangent to the end of the sphere across the entire opening. After converting the flux to a molecule rate (#/s) and allowing for the ratio between simulated and real particles the number of molecules per time step is calculated. The position of each molecule is chosen based on a random homogenous distribution on a circle, and then the coordinates are translated to the entrance. The velocity components are selected by sampling from the same distribution as was used in diffuse reflection off the surface, where a velocity weighted Maxwell-Boltzmann distribution was utilized, along with the appropriate vector transformations. Sample flux distributions are shown in Figure 11 from viewpoints tangential, and normal to the entrance.

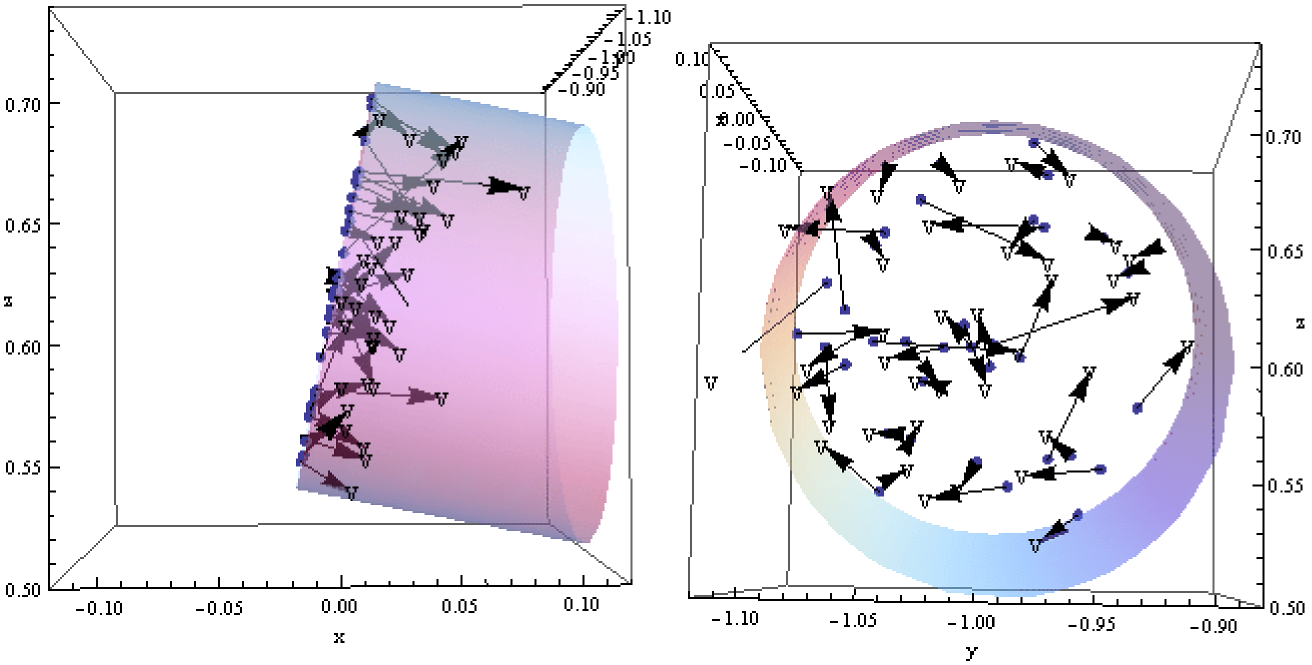


Figure 10. Velocity vectors of individual molecules corresponding to the source flux during an individual time step.

The parameters of the system we simulated are presented in Table 2.

Table 2. Simulation parameters.

Source Pressure (Pa)	0.5
Initial Pressure (Pa)	0.0
Temperature of Walls and Source Gas (K)	273.15
Real / Simulated Particle Ratio	3.43469×10^{10}
Mean Free Path (m)	0.0126866
Tube Radius (m)	0.0126866 = 1 mfp
Tube Length (radians)	π/2
Tube Length (m)	0.126866 = 10 mfp
Tube Pitch	0.2
Simulation Time (s)	2.5×10^{-3}
Maximum Number of Clusters	30
Number of Runs	10

The tube we used in this set of simulations is shown below in Figure 12.

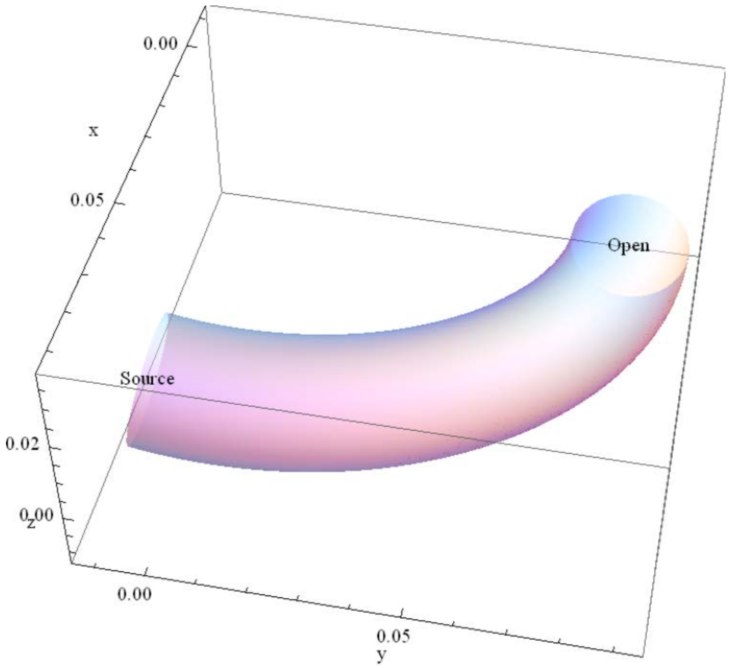
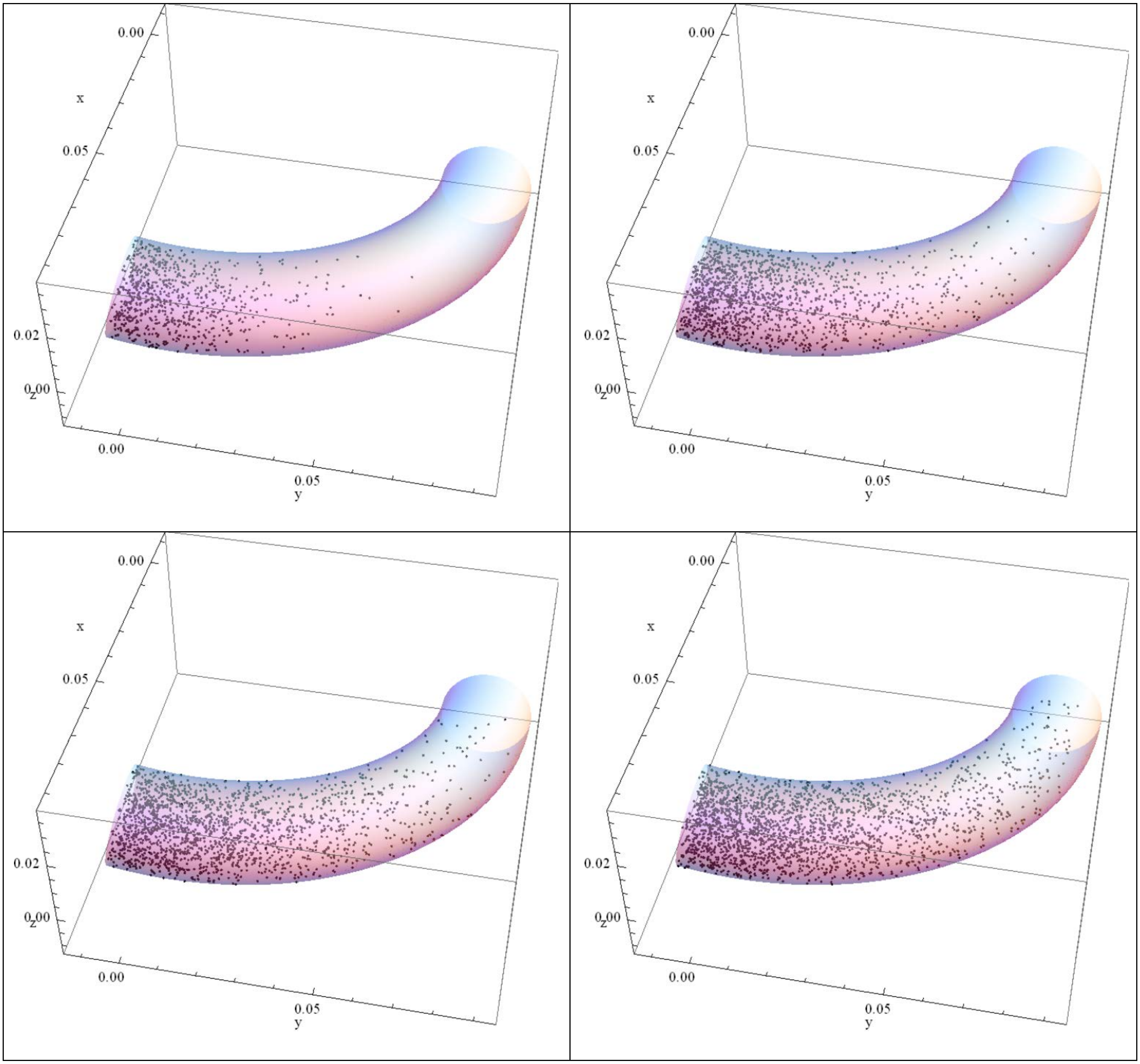
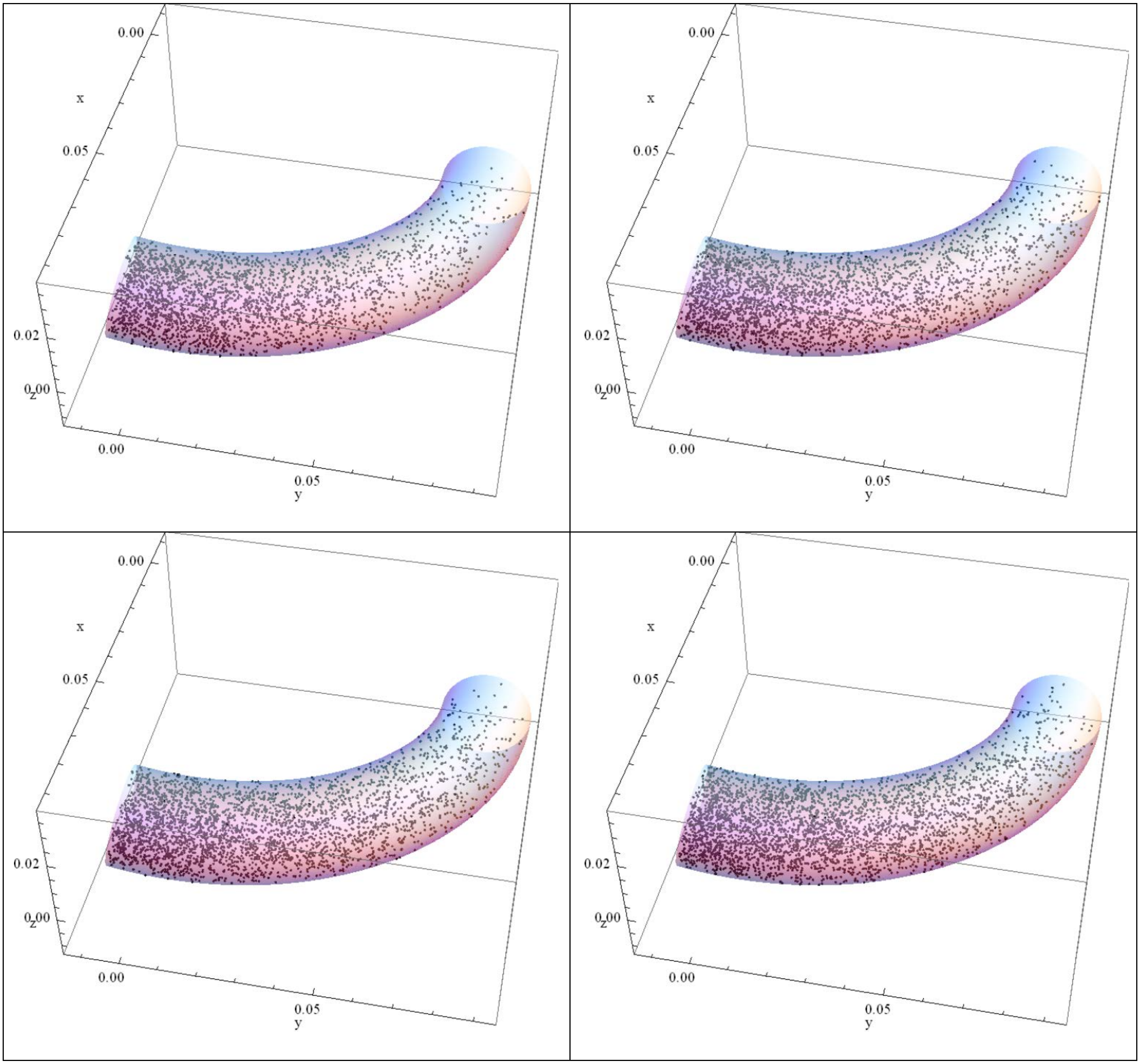


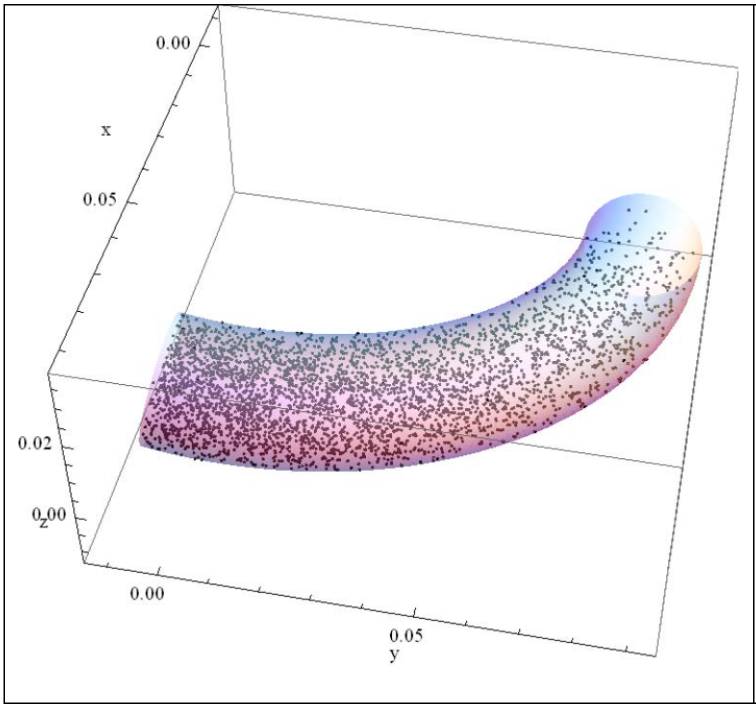
Figure 11. Geometry of simulation tube.

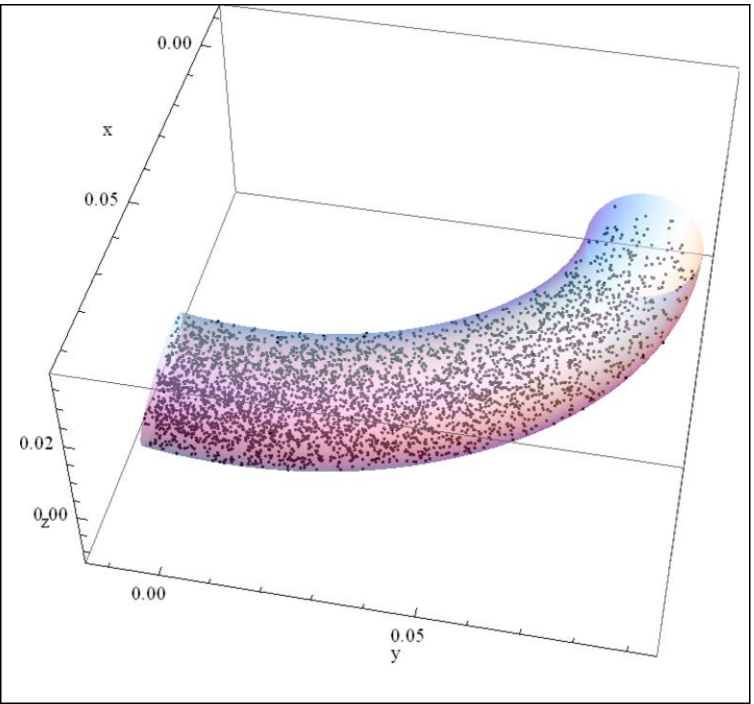
Several early time steps are illustrated in Table 3 where the leading edge of molecules travelling down the tube can be clearly seen. The simulation is run further than the graphs illustrate, but it is difficult to visually resolve the difference at that point.

Table 3. Movement of gas particles from 1.25 × 10^{-4} (s) to 1.25 × 10^{-3} (s) in steps of 1.25 × 10^{-4} (s).




0.00
x
0.05
0.02
0.00
z
0.00
0.05
y
0.00
x
0.05
0.02
0.00
z
0.00
0.05
y
0.00
x
0.05
0.02
0.00
z
0.00
0.05
y
0.00
x
0.05
0.02
0.00
z
0.00
0.05
y





A more objective presentation of the results obtained is presented in the following sequence of figures. Figure 13 through Figure 16 show the number of simulated particles per equidistant slice of the helix along the parametric curve. Note that the source is on the right hand side of the plot. The allocation of particles based on their Cartesian coordinates to a specific slice is accomplished by utilizing the same root-finding method used in the simulation to determine the closest part of the parametric curve with respect to the particle. This has been found to be computationally intensive with regard to results analysis. In the future this information will be saved during the simulation to eliminate re-computation during the results analysis.

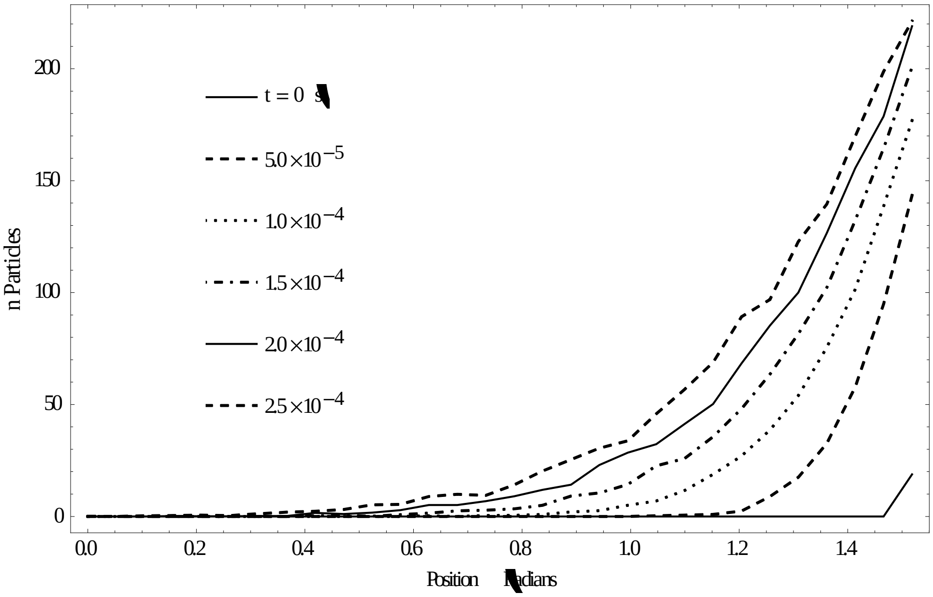


Figure 12. Number of particles in the tube (1 of 4).

In Figure 13 there is an initial rapid increase in the number of particles in the source end of the tube, followed by a deceleration of the increase. This corresponds to the initial filling of the slice by the source, followed by particles beginning to interact with the walls and each other. This results in the particles either exiting the tube at the source, or continuing to travel down the tube, increasing the number count in further slices.

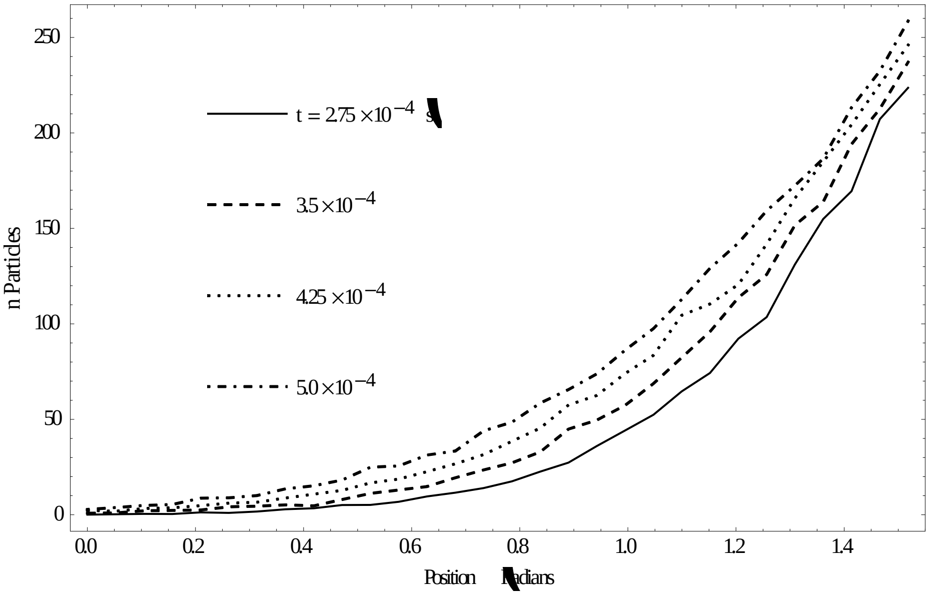


Figure 13. Number of particles in the tube (2 of 4).

There is only slight additional growth in the source slice in Figure 14 as it begins to reach equilibrium. However, there is continued growth through the rest of the tube, as particles begin to reach the open end of the tube.

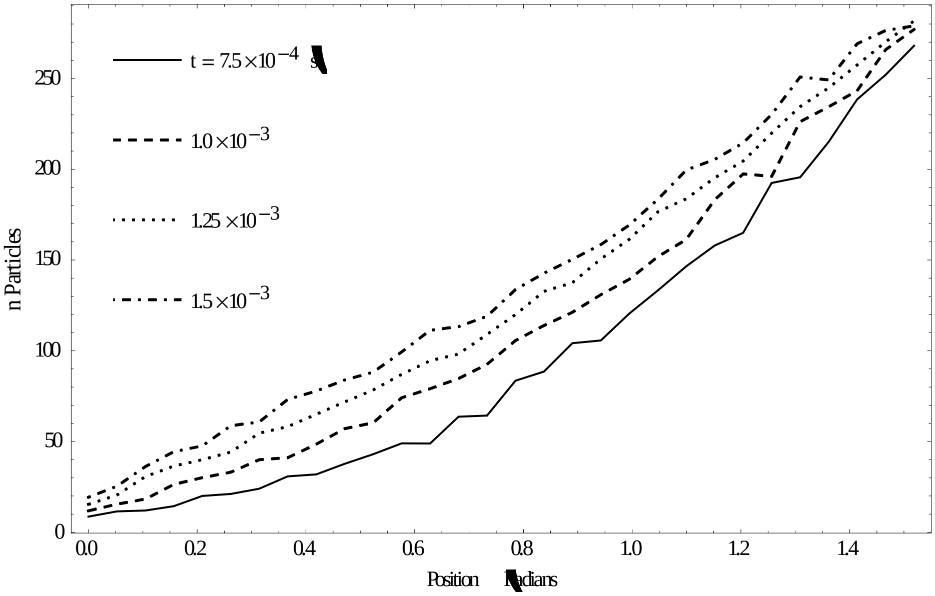


Figure 14. Number of particles in the tube (3 of 4).

Figure 15 and Figure 16 show that as the distribution begins to reach a steady state condition that the number concentration approaches a linear distribution as expected for this type of problem.

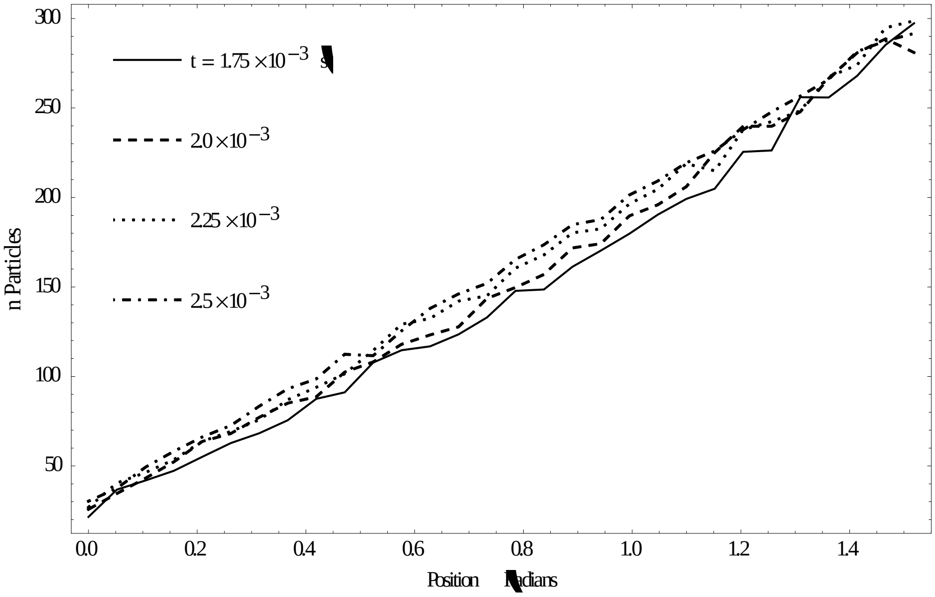


Figure 15. Number of particles in the tube (4 of 4).

Figure 17 through Figure 20 show the distribution of the temperature of the gas at different times. It is important to note that the high degree of noise in the temperature data is not meaningful, but is an artifact of the small number of particles present in some of the slices. As the gas and the walls are the same average temperature, the equilibrium behavior is not of much interest; however the transient behavior has several characteristics to note. The temperature at the source end quick stabilizes around the expected mean temperature, but the leading edge of the particle flow generally has a much lower temperature even though it is expected that these particles would have a higher energy in order to have travelled farther than the majority of particles. This is because the majority of their energy is in bulk kinetic flow, with little deviation from the average to produce thermal energy. This can be seen from plotting the bulk velocity at the end of the simulation, which is shown in Figure 21.

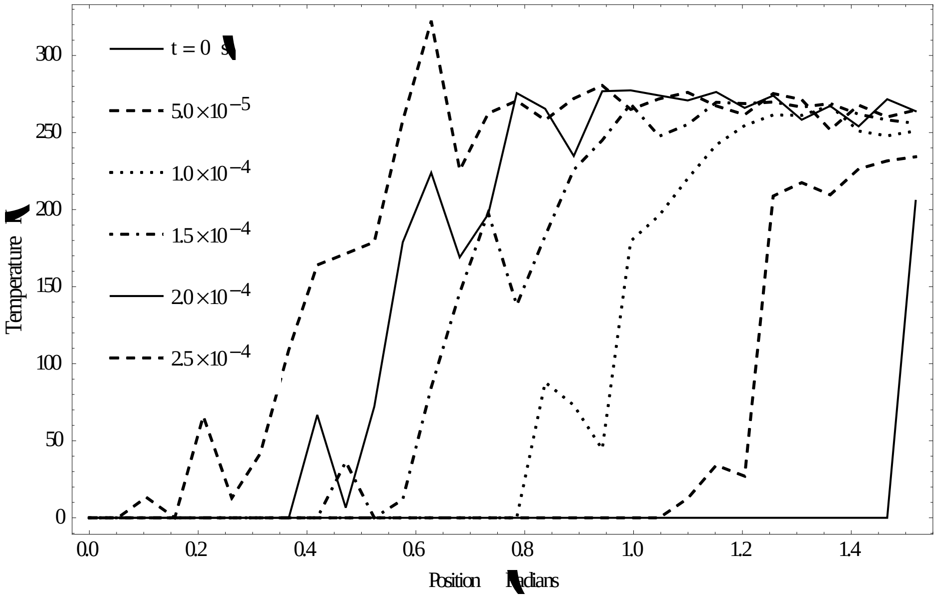


Figure 16. Temperature of the gas in the tube (1 of 4).

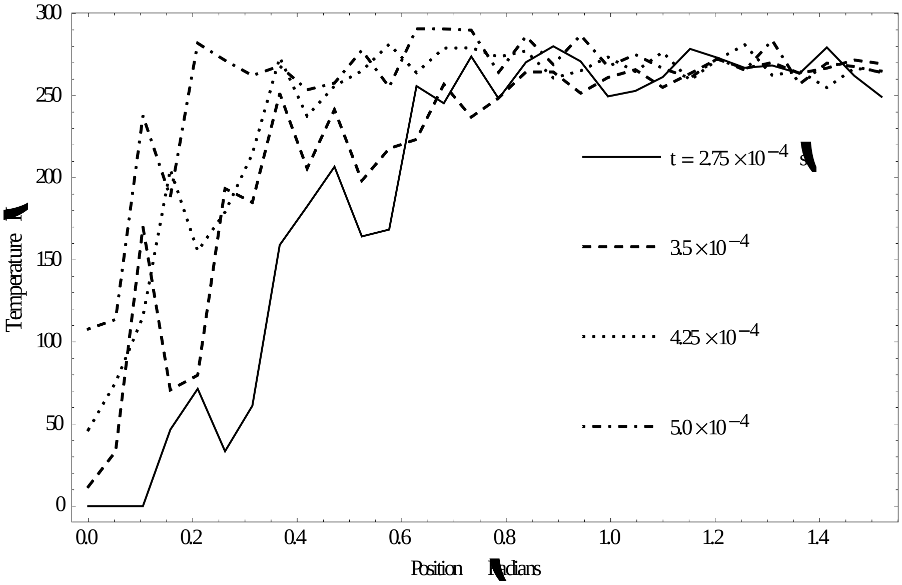


Figure 17. Temperature of the gas in the tube (2 of 4).

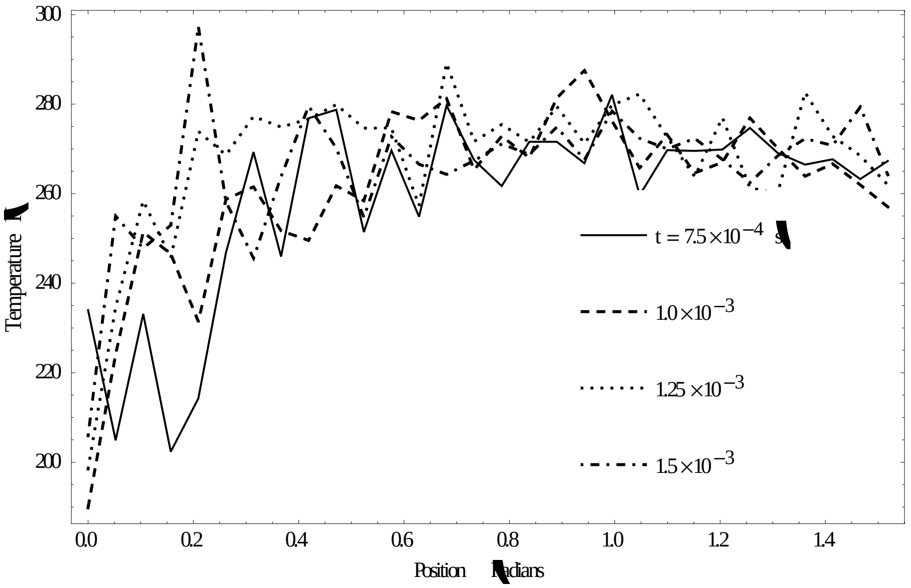


Figure 18. Temperature of the gas in the tube (3 of 4).

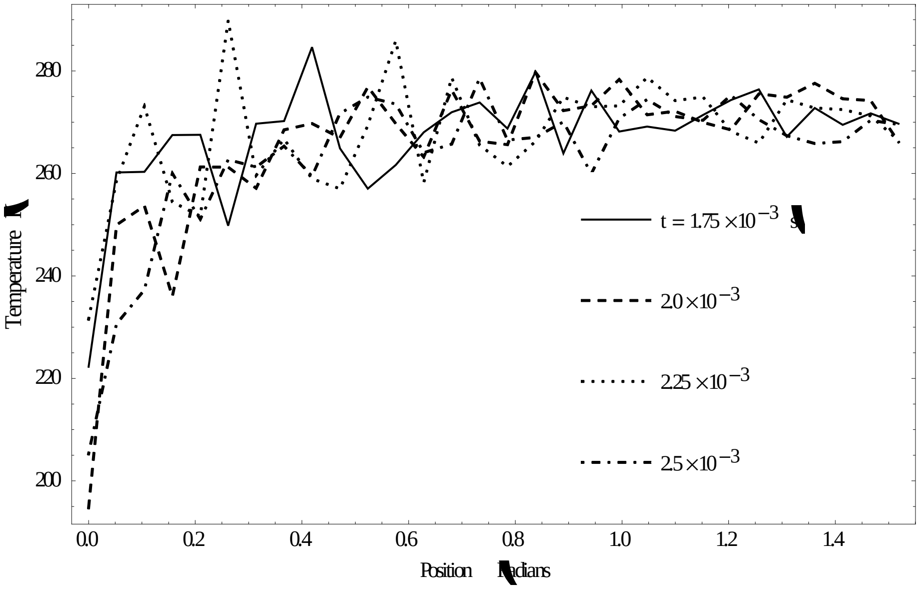


Figure 19. Temperature of the gas in the tube (4 of 4).

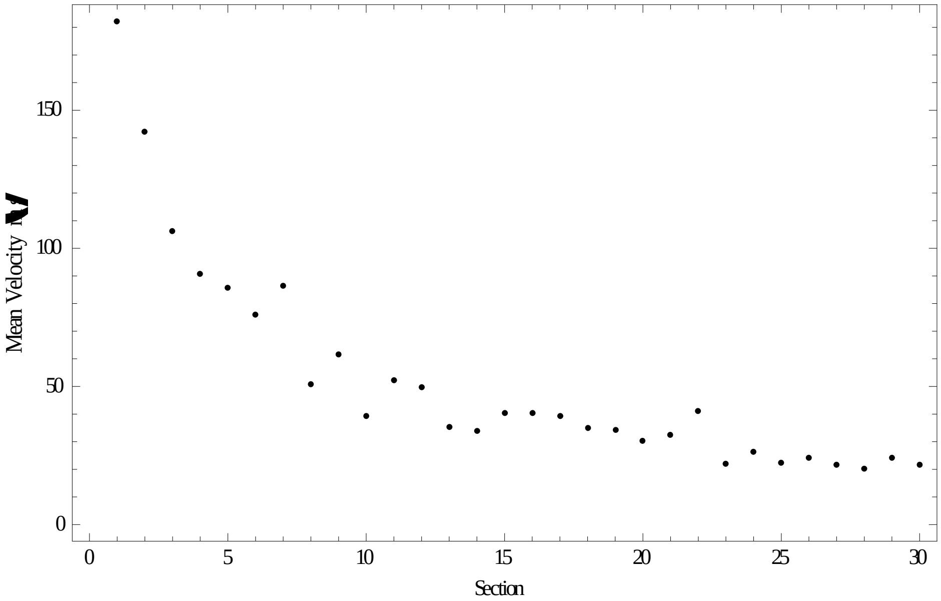


Figure 20. Bulk velocity distribution by section at t = 2.5 × 10^{-3} (s).

Figure 22 through Figure 25 show the pressure of the gas over time, with nearly the same curve development behavior as shown in the number distribution. There is some noise introduced by the dependence of pressure upon temperature, but otherwise the comments made regarding the number distribution apply here as well.

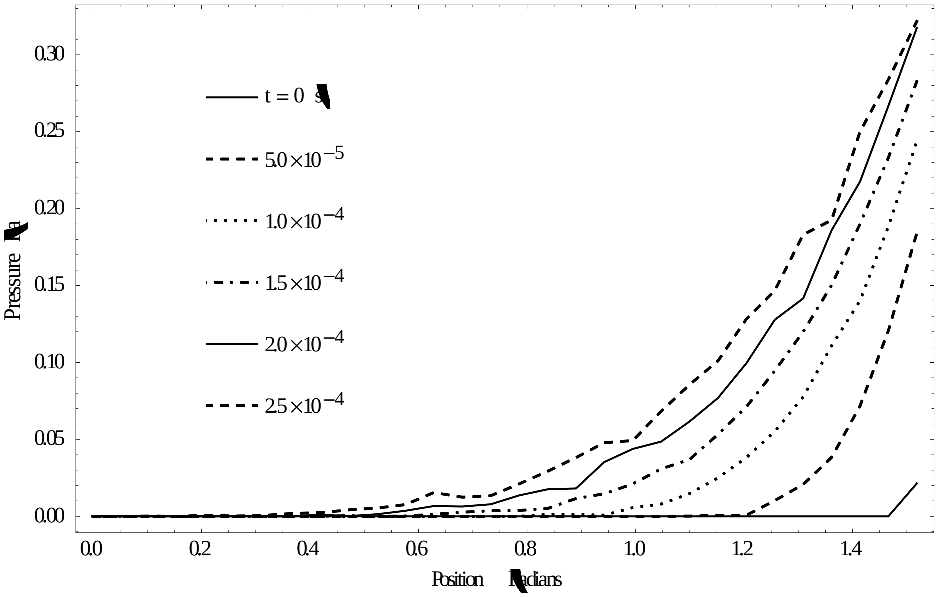


Figure 21. Pressure of the gas in the tube (1 of 4).

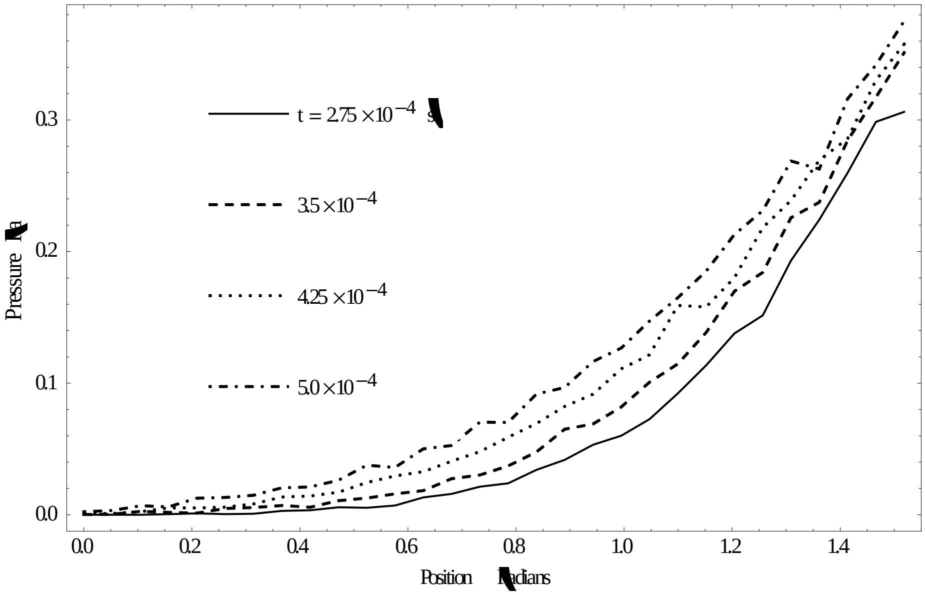


Figure 22. Pressure of the gas in the tube (2 of 4).

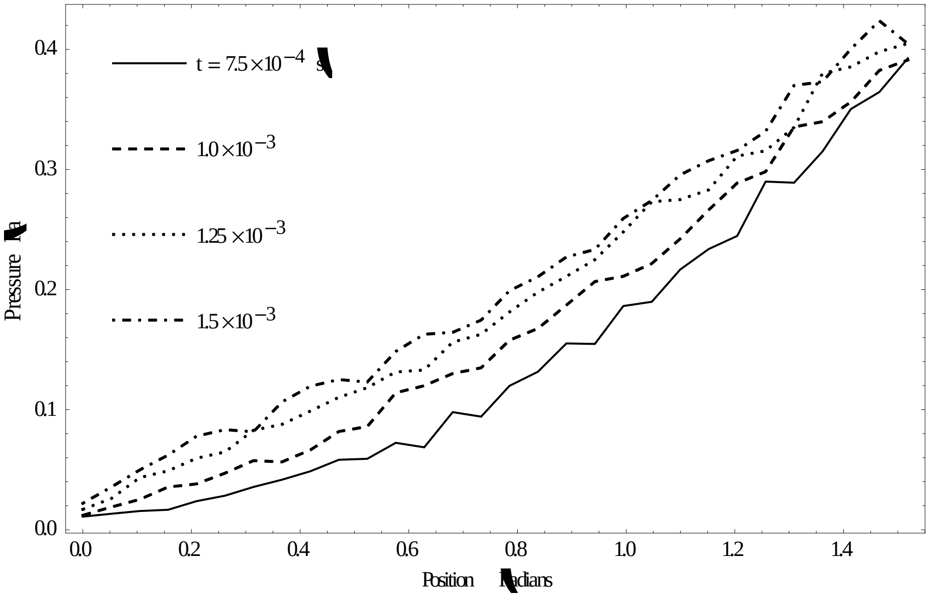


Figure 23. Pressure of the gas in the tube (3 of 4).

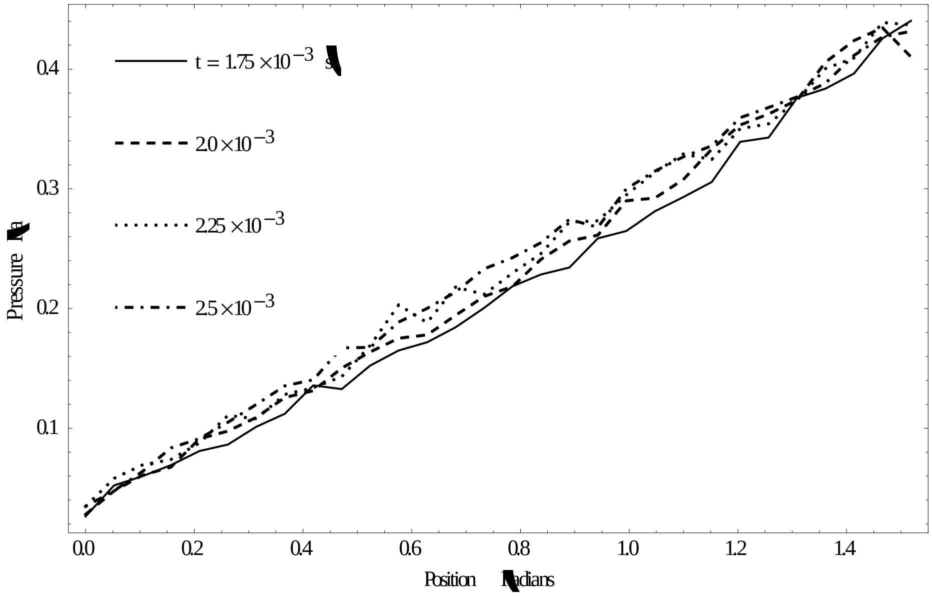
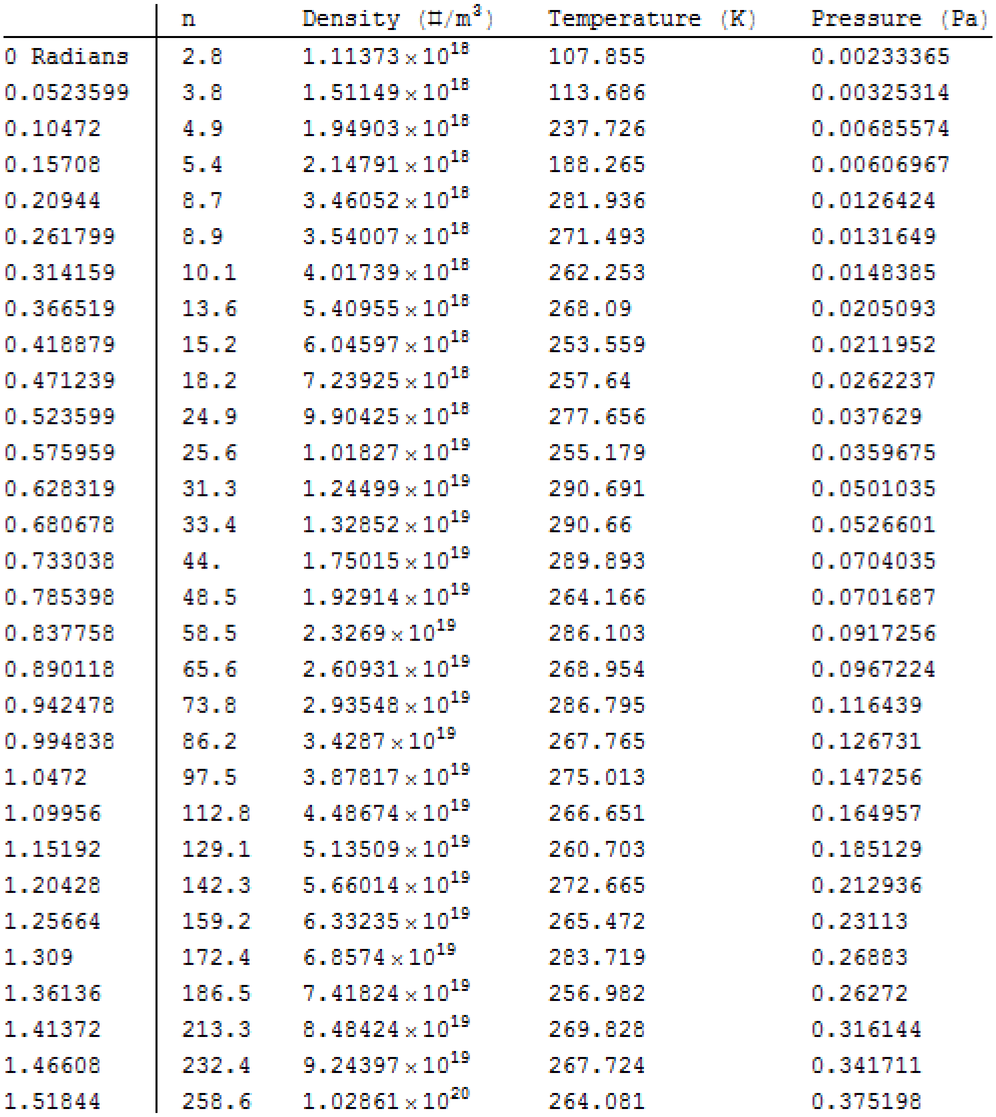


Figure 24. Pressure of the gas in the tube (4 of 4).

Table 4 and Table 5 are a sample of data sets used in generating the earlier figures. These particular data sets represent 20% and 100% completion of the simulation. The primary points to note are that between the two tables the value of *n* for the source slice increased only 12.8%, while the open slice value of *n* increased 982%. So that while relatively early in the simulation one end of the tube is almost at equilibrium, while the other end is in a very transient state.

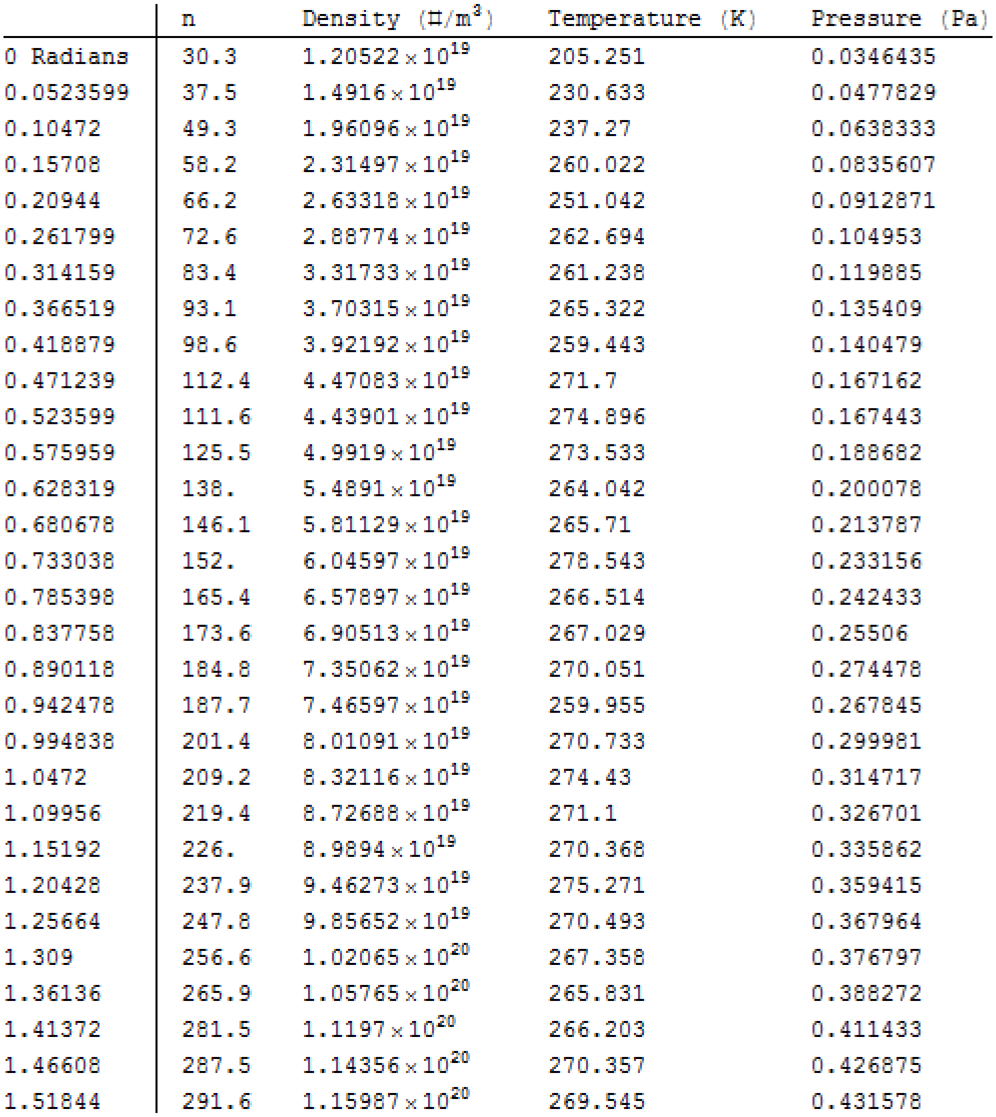
Table 4. Mean value of data at t = 5 × 10^{-4} (s).



	n	Density (#/m^3)	Temperature (K)	Pressure (Pa)
0 Radians	2.8	1.11373×10^{18}	107.855	0.00233365
0.0523599	3.8	1.51149×10^{18}	113.686	0.00325314
0.10472	4.9	1.94903×10^{18}	237.726	0.00685574
0.15708	5.4	2.14791×10^{18}	188.265	0.00606967
0.20944	8.7	3.46052×10^{18}	281.936	0.0126424
0.261799	8.9	3.54007×10^{18}	271.493	0.0131649
0.314159	10.1	4.01739×10^{18}	262.253	0.0148385
0.366519	13.6	5.40955×10^{18}	268.09	0.0205093
0.418879	15.2	6.04597×10^{18}	253.559	0.0211952
0.471239	18.2	7.23925×10^{18}	257.64	0.0262237
0.523599	24.9	9.90425×10^{18}	277.656	0.037629
0.575959	25.6	1.01827×10^{19}	255.179	0.0359675
0.628319	31.3	1.24499×10^{19}	290.691	0.0501035
0.680678	33.4	1.32852×10^{19}	290.66	0.0526601
0.733038	44.	1.75015×10^{19}	289.893	0.0704035
0.785398	48.5	1.92914×10^{19}	264.166	0.0701687
0.837758	58.5	2.3269×10^{19}	286.103	0.0917256
0.890118	65.6	2.60931×10^{19}	268.954	0.0967224
0.942478	73.8	2.93548×10^{19}	286.795	0.116439
0.994838	86.2	3.4287×10^{19}	267.765	0.126731
1.0472	97.5	3.87817×10^{19}	275.013	0.147256
1.09956	112.8	4.48674×10^{19}	266.651	0.164957
1.15192	129.1	5.13509×10^{19}	260.703	0.185129
1.20428	142.3	5.66014×10^{19}	272.665	0.212936
1.25664	159.2	6.33235×10^{19}	265.472	0.23113
1.309	172.4	6.8574×10^{19}	283.719	0.26883
1.36136	186.5	7.41824×10^{19}	256.982	0.26272
1.41372	213.3	8.48424×10^{19}	269.828	0.316144
1.46608	232.4	9.24397×10^{19}	267.724	0.341711
1.51844	258.6	1.02861×10^{20}	264.081	0.375198

The final pressure in the source section is 0.432 Pa, which is approaching the source pressure of 0.5 Pa. This difference is most likely due to full equilibrium not having been reached yet, fluctuations due to Monte Carlo, and that the pressure was spatially averaged.

Table 5. Mean value of data at t = 2.5 × 10^{-3} (s).



	n	Density (#/m^3)	Temperature (K)	Pressure (Pa)
0 Radians	30.3	1.20522×10^{19}	205.251	0.0346435
0.0523599	37.5	1.4916×10^{19}	230.633	0.0477829
0.10472	49.3	1.96096×10^{19}	237.27	0.0638333
0.15708	58.2	2.31497×10^{19}	260.022	0.0835607
0.20944	66.2	2.63318×10^{19}	251.042	0.0912871
0.261799	72.6	2.88774×10^{19}	262.694	0.104953
0.314159	83.4	3.31733×10^{19}	261.238	0.119885
0.366519	93.1	3.70315×10^{19}	265.322	0.135409
0.418879	98.6	3.92192×10^{19}	259.443	0.140479
0.471239	112.4	4.47083×10^{19}	271.7	0.167162
0.523599	111.6	4.43901×10^{19}	274.896	0.167443
0.575959	125.5	4.9919×10^{19}	273.533	0.188682
0.628319	138.	5.4891×10^{19}	264.042	0.200078
0.680678	146.1	5.81129×10^{19}	265.71	0.213787
0.733038	152.	6.04597×10^{19}	278.543	0.233156
0.785398	165.4	6.57897×10^{19}	266.514	0.242433
0.837758	173.6	6.90513×10^{19}	267.029	0.25506
0.890118	184.8	7.35062×10^{19}	270.051	0.274478
0.942478	187.7	7.46597×10^{19}	259.955	0.267845
0.994838	201.4	8.01091×10^{19}	270.733	0.299981
1.0472	209.2	8.32116×10^{19}	274.43	0.314717
1.09956	219.4	8.72688×10^{19}	271.1	0.326701
1.15192	226.	8.9894×10^{19}	270.368	0.335862
1.20428	237.9	9.46273×10^{19}	275.271	0.359415
1.25664	247.8	9.85652×10^{19}	270.493	0.367964
1.309	256.6	1.02065×10^{20}	267.358	0.376797
1.36136	265.9	1.05765×10^{20}	265.831	0.388272
1.41372	281.5	1.1197×10^{20}	266.203	0.411433
1.46608	287.5	1.14356×10^{20}	270.357	0.426875
1.51844	291.6	1.15987×10^{20}	269.545	0.431578

Finally we looked at the net flow of particles at each end of the tube during the course of the simulation. For this simulation the particle source was 18.7491 particles per time step, therefore any particles lost from the source end of the tube are subtracted from this value when calculating the net flow. The particle net is shown in Figure 26 where positive values represent particles entering the tube, and negative values particles exiting. Initially the net source and net values are the same as no particles are lost, but over time the net begins to approach zero. The noise in the distribution can make this difficult to visualize, therefore the cumulative distribution is shown in Figure 27, where the slope of the net line begins to approach, but has not yet reached zero. Therefore the simulation has not completely reached equilibrium and a slighter longer runtime would be appropriate in this case. From initial testing we have determined that the simulation needs to run between 50-100% longer to completely reach steady state conditions, but the change that occurs over that time is minimal compared to the results presented in this report.

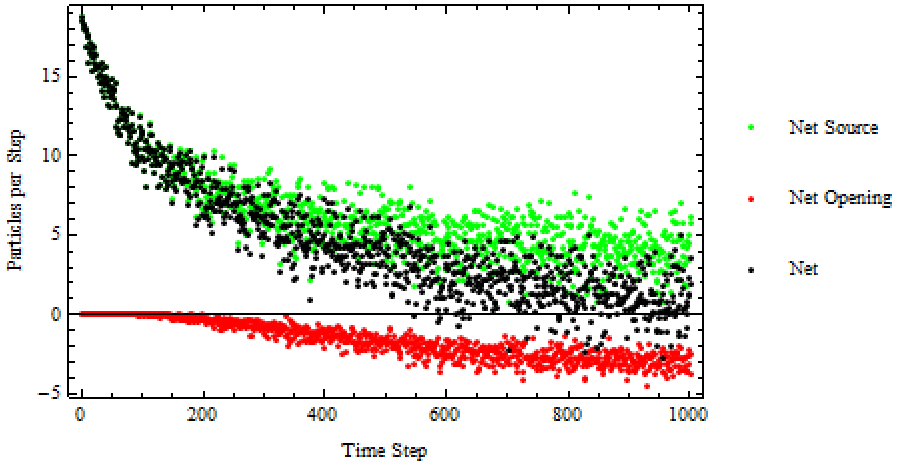


Figure 25. Net gain and loss of molecules at each end of the tube per time step.

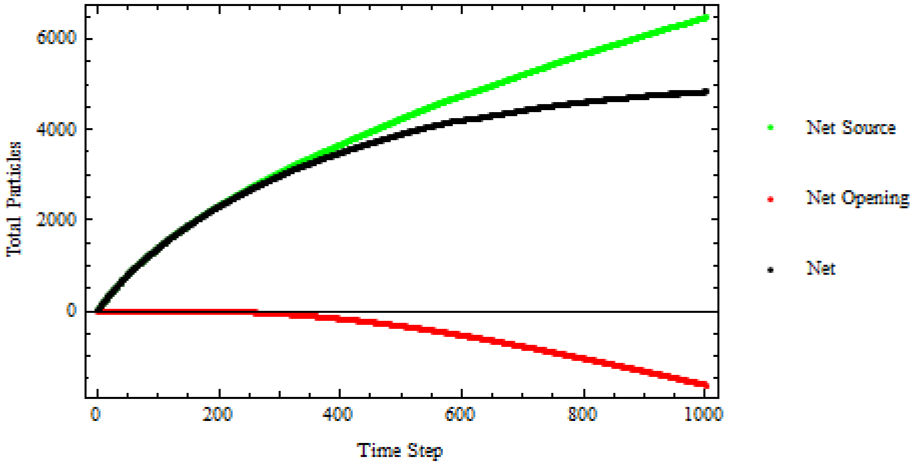


Figure 26. **Cumulative gain and loss of molecules at each end of the tube per time step.**

Through the course of this work we encountered several instances where our results varied substantially from the numerical method used for benchmarking. While this is to be expected in complex systems, in the simple test cases we were considering further evaluation is necessary to determine if the difference is reasonable, or if it is due to implementation errors or programming bugs.

We have been perplexed by differences in values of the heat flux between the q-moments method and our DSMC program as the Knudsen number (Kn) becomes smaller. For small Kn We suspect an implementation error in our program. Varying the volume of the simulation volume has not resulted in a change in results, so it is not an issue of simulated number density. We have also increased the number of cells/clusters considered to determine if that was affecting the simulation and our results did not change.

We considered several different methods for defining molecule groups. The benchmark cell discretization method over the x-axis and discretizing by the clustering method over the x-axis both led to comparable results. However when clustering over all space the result was a flattened temperature profile and a smaller heat flux relative to the other two methods. Decreasing the volume of the simulation over the axes perpendicular to heat transfer had no effect on the results. Increasing the number of clusters has not noticeably changed to results.

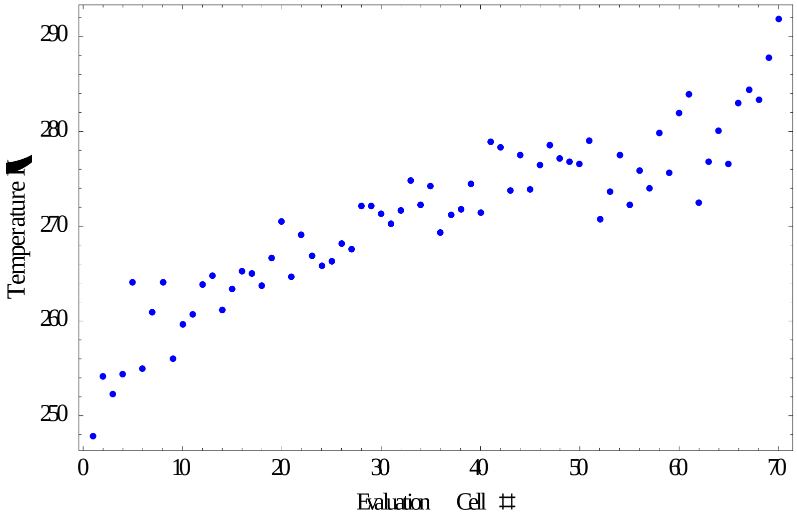


Figure 27. Temperature of infinite parallel plate system with Kn=0.1 and 70 clusters and result cells.

For the future, we will evaluate the effect of the number of simulation particles on the results and will work to accelerate the run time of the simulations to allow an increased number of particles to be simulated. We are also looking at our implementation of the clustering volume and its impact on the particle collision rate.

Previously the molecules in the parallel plate simulation were allowed to travel outside the control volume while still being counted to maintain the correct number density. As the simulation is homogeneous it did not affect the cell based simulation. However this does affect how the volume of the particle clusters is calculated and so now when a particle reaches the edge of the volume it is reflected back into the control volume, representing a particle from outside flowing into the control volume.

To this point we have been using the native clustering algorithm in Mathematica. However it has limitations due to our adaptation of it from its original purpose. This algorithm operates on the premise that there is no prior knowledge about the system. While this is true when the system is initialized, after the first clustering event information does exist but the native algorithm does not preserve it for future use. Thus we have begun working on our own implementation which will permit the retention of past information to provide a starting point for the next clustering event. As each molecule only moves a small distance in a time step it will not change its position significantly with respect to its previous cluster. Therefore we can use the prior center molecule as the initial starting point instead of using random selection and iterating each clustering event until a suitable geometry is selected. This will also allow us to limit the search for a more favorable geometry to those molecules nearest the center molecule without having to test the cluster's entire inventory of molecules. In addition this will permit us to significantly increase the number of clusters in a simulation as the processing time will only increase by between the order of N and $N \log N$ versus N^2 using the previous clustering technique.

Nuclear Science and Engineering

Computation of Aerosol Evolution under Coagulation

--Manuscript Draft--

Manuscript Number:	NSE14-91R1
Full Title:	Computation of Aerosol Evolution under Coagulation
Article Type:	Technical Paper
Keywords:	computation, aerosols, coagulation
Corresponding Author:	Sudarshan K Loyalka, PhD University of Missouri Columbia, MO UNITED STATES
Corresponding Author Secondary Information:	
Corresponding Author's Institution:	University of Missouri
Corresponding Author E-Mail:	loyalkas@missouri.edu
Corresponding Author's Secondary Institution:	
First Author:	Shawn A Campbell, MS
First Author Secondary Information:	
Order of Authors:	Shawn A Campbell, MS
	Sudarshan K Loyalka, PhD
Order of Authors Secondary Information:	
Manuscript Region of Origin:	UNITED STATES
Abstract:	Understanding and improved modeling of aerosol evolution in nuclear reactor accidents is important in estimations of the Nuclear Source Term. We explore here the nature of some approximations inherent in the widely used sectional technique for both single and multi-component aerosols and the influence these have on results. We also describe our efforts towards improving the fidelity of the sectional technique to the actual physics by coupling the sectional technique with the Direct Simulation Monte Carlo simulations, and why such coupling has proved difficult.
Suggested Reviewers:	
Author Comments:	

Computation of Aerosol Evolution under Coagulation

Shawn A. Campbell & Sudarshan K. Loyalka*

Nuclear Science and Engineering Institute
University of Missouri
E2433 Lafferre Hall
Columbia, MO 65211

*Corresponding author
Email: LoyalkaS@missouri.edu
Phone: 573-882-3568
Fax: 573-884-4801

No. of Pages:	54
No. of Tables:	10
No. of Figures:	15

ABSTRACT

Understanding and improved modeling of aerosol evolution in nuclear reactor accidents is important in estimations of the Nuclear Source Term. We explore here the nature of some approximations inherent in the widely used sectional technique for both single and multi-component aerosols and the influence these have on results. We also describe our efforts towards improving the fidelity of the sectional technique to the actual physics by coupling the sectional technique with the Direct Simulation Monte Carlo simulations, and why such coupling has proved difficult.

I. **Introduction**

Understanding and improved modeling of aerosol evolution in nuclear reactor accidents is important in estimations of Nuclear Source Term.[1-4] The sectional technique as embodied in the MAEROS program is employed by source term codes such as CONTAIN[1] and MELCOR[2] to model multicomponent aerosol particles within a reactor vessel and containment. This technique is used also in some CFD codes where aerosol loadings are large, and coagulation is significant. While this technique has a great advantage in its computational speed, it has been shown[5-8] to be insufficient in some cases of multi-component aerosols due to its algorithmic limitations requiring averaging over number and component densities in its foundational approximations.

In contrast, the Direct Simulation Monte Carlo (DSMC) method has been shown[5-8] to accurately model multi-component aerosols without the need for such approximations. The method uses a sample of particles of various masses and component distributions from the initial distribution and evolves this sample in time accounting for numerous rate processes. Generally, a time step is chosen, and a number are sampled for collisions according to various algorithms. While DSMC maintains fidelity to aerosol physics for particle interactions, its computation time is often orders of magnitude greater than that of the sectional method in modeling the same scenario. We note that, while this present work advances the previous work of our group, others have applied Monte Carlo methods to modeling aerosol evolution as well.[10-13]

Our purpose in this paper is to clarify the approximations made by the sectional technique and the impact they can have on the accuracy of a simulation. We also describe our explorations towards coupling the DSMC and sectional methods. In Section II, we give a brief description of the sectional and DSMC techniques. In Section III we benchmark both the sectional and DSMC techniques by comparing them to a single component exact solution. Section IV contains detailed explanation and analysis of the sectional approximations on the number distribution of a single component aerosol. An exact solution for a multicomponent aerosol is used to benchmark the sectional and DSMC simulations in Section V. In Section VI we describe in greater detail assumptions used in the sectional approximation on the component distribution for a multicomponent aerosol. We then describe our explorations towards coupling the sectional and DSMC methods in Section VII. In Section VIII we briefly outline recent improvements that we have made to improve computational speed of the DSMC program. Finally, we present our conclusions in Section IX.

It is important to note that for computations here we have not used MAEROS per se. Rather, we have used our own in house Mathematica program wherein we have implemented the sectional method (we have verified that the results of this program compare well with those of MAEROS).

II. Previous Works

The general dynamic equation for the coagulation of an aerosol is well-established.[14,15] We refer here, in particular, to Williams and Loyalka[3] and their derivation. Considering a homogeneous system of spherical aerosols comprised of N components, we wish to determine the number of particles with volume in the range v_p to $v_p + dv_p$ and mass from m_p to $m_p + dm_p$ of a given component p, given by the aerosol distribution function

$$n(v_1, v_2, \ldots, v_N; m_1, m_2, \ldots m_N, t)\ dv_1\ dv_2 \ldots dv_N\ dm_1\ dm_2 \ldots dm_N \qquad (1)$$

The total volume and mass of the particle will then be given by

$$v = \sum_{p=1}^{N} v_p \tag{2}$$

and

$$m = \sum_{p=1}^{N} m_p \tag{3}$$

respectively. Making use of the shorthand notation $n(\boldsymbol{v}, \boldsymbol{m}, t) = n(v_1, v_2, \ldots, v_N; m_1, m_2, \ldots m_N, t)$ and $d\boldsymbol{v} = dv_1 dv_2 \ldots dv_N$ the general dynamic equation can be expressed as

$$\begin{aligned}
&\frac{\partial}{\partial t} n(\boldsymbol{v}, \boldsymbol{m}, t) + R(\boldsymbol{v}, \boldsymbol{m}, t)\, n(\boldsymbol{v}, \boldsymbol{m}, t) \\
&\qquad + \sum_{p=1}^{N} \frac{\partial}{\partial v_p} \left[I_p(\boldsymbol{v}, \boldsymbol{m}, t)\, n(\boldsymbol{v}, \boldsymbol{m}, t) \right] \\
&\qquad = \frac{1}{2} \int_0^\infty d\boldsymbol{v} \int_0^\infty d\boldsymbol{w} \int_0^\infty d\boldsymbol{q} \int_0^\infty d\boldsymbol{s} \times n(\boldsymbol{u}, \boldsymbol{q}, t)\, n(\boldsymbol{w}, \boldsymbol{s}, t)\, K(u, q | w, s) \\
&\qquad \times \prod_{p=1}^{N} \delta(v_p - u_p - w_p)\, \delta(m_p - q_p - s_p) \\
&\qquad - n(\boldsymbol{v}, \boldsymbol{m}, t) \int_0^\infty d\boldsymbol{u} \int_0^\infty d\boldsymbol{q}\, K(u, q | v, m)\, n(\boldsymbol{u}, \boldsymbol{q}, t) + S(\boldsymbol{v}, \boldsymbol{m}, t)
\end{aligned} \tag{4}$$

where $q = \sum_{p=1}^{N} q_p$ and $s = \sum_{p=1}^{N} s_p$. The function $K(u, q|v, m)$ refers to the coagulation coefficient of two particles with volumes u and v and masses q and m respectively (we will later use the notation β also to denote this), $R(\boldsymbol{v}, \boldsymbol{m}, t)$ is the rate of removal due to settling and deposition, $I_p(\boldsymbol{v}, \boldsymbol{m}, t)$ represents the condensation rate, and $S(\boldsymbol{v}, \boldsymbol{m}, t)$ is a source term. For a detailed explanation of the terms involved, the reader is referred to Williams and Loyalka.[3] For coagulation alone, which is the focus of this work, this equation becomes:

$$\begin{aligned}\frac{\partial}{\partial t} n(\boldsymbol{v},\boldsymbol{m},t) = \frac{1}{2}\int_0^\infty d\boldsymbol{v}\int_0^\infty d\boldsymbol{w}\int_0^\infty d\boldsymbol{q}\int_0^\infty d\boldsymbol{s} \times n(\boldsymbol{u},\boldsymbol{q},t)\, n(\boldsymbol{w},\boldsymbol{s},t)\, K(u,q|w,s) \\ \times \prod_{p=1}^{N} \delta\left(v_p - u_p - w_p\right)\delta\left(m_p - q_p - s_p\right) \\ - n(\boldsymbol{v},\boldsymbol{m},t)\int_0^\infty d\boldsymbol{u}\int_0^\infty d\boldsymbol{q}\, K(u,q|v,m)\, n(\boldsymbol{u},\boldsymbol{q},t)\end{aligned}$$

II.A. Description of Sectional Technique

The sectional technique was designed by Gelbard and Seinfeld[16-18] as a means for modeling multicomponent particle interaction. At the core of this technique is the conservation of some size variable, v, (could be number or volume but is typically mass) for each of s components over m sections (size bins) during the processes of coagulation, condensation, particle source, removal, and intraparticle chemical reactions. For the purposes of this paper, attention has been placed solely on aerosol coagulation and an explanation of the multicomponent sectional equation for coagulation is provided below.

To clarify the operational parts of the technique, one begins by discretizing the range of particle diameters $(d_{min},\ d_{max})$ into m subdivisions. These subdivisions are assumed to be geometrically spaced such that the lower and upper bounds of section ℓ is given by

$$\left(d_{min}\left(\frac{d_{max}}{d_{min}}\right)^{\frac{\ell-1}{m}}, d_{min}\left(\frac{d_{max}}{d_{min}}\right)^{\frac{\ell}{m}}\right). \tag{4}$$

This geometrical division is not essential, but is convenient for computational speed of certain coagulation rate constants as used in the technique. Hence a particle will belong to some corresponding volume group $\ell \in \{1,2,\dots,m\}$ with mass in the range $[v_{\ell-1}, v_\ell]$ since, as discussed later, each particle is assumed to have a constant mass density regardless of composition. Consequently, for simulations within this work, a density of

$$\rho = 1000\ kg/m^3$$

is used for all aerosol particles of all component types.

For a single component aerosol, the aerosol distribution function will be of the form $n(v,t)$ where $n(v,t)dv$ is the number concentration of particles in the range $[v, v + dv]$ at time t. From this we define a new function $Q_\ell(t)$ as the total mass of aerosol per unit volume in section ℓ at time t given by

$$Q_\ell(t) = \int_{v_{\ell-1}}^{v_\ell} v \cdot n(v,t)dv. \tag{5}$$

With a multicomponent aerosol, however, one must instead monitor the coagulation of each component into and out of each section. Since all components are assumed to have the same mass density, in each section one ignores the component specifity, and assumes that the intrasectional aerosol distribution function can be specified as $n(v,t)\ dv$. In this case, one defines, $Q_{\ell,k}(t)$ as the mass density (mass per unit volume) of component k in section ℓ so that

$$Q_\ell(t) = \sum_{k=1}^{s} Q_{\ell,k}\,(t) = \int_{v_{\ell-1}}^{v_\ell} v \cdot n(v,t)dv. \tag{6}$$

Also, $\bar{v}_{\ell,k}$ is defined to be the average mass of component, k, in the mass range $[v, v + dv]$.

The function $\beta(u,v)$ is defined as the number of coagulations per unit time between particles of sizes u and v due to turbulent, gravitational and Brownian influences. The reader is referred to Williams and Loyalka[3] for a detailed a description of this coagulation kernel. The rate of coagulation is then given by $\beta(u,v)n(u,t)n(v,t)du\ dv$ for particles in the size ranges $[u, u+du]$ and $[v, v+dv]$.

In order to create a system of ordinary differential equations, Gelbard and Seinfeld[16] make two approximations to relate both $n(v,t)$ and $\bar{v}_{\ell,k}$ to $Q_{\ell,k}$. To relate $n(v,t)$, the assumption is made that all particles within a given section have the same size distribution. This leads to the approximation that $n(v,t) = \frac{Q_\ell f'(v)}{v(x_\ell - x_{\ell-1})}$ where $x = f(v)$ is a conveniently chosen size variable kept constant within each section. The second approximation is to assume that the mean component mass fraction of all particles in each section is equal (but it is distinct from section to section). The resultant relationship between Q_ℓ and $\bar{v}_{\ell,k}$ is

$$\bar{v}_{\ell,k} = \frac{vQ_{\ell,k}}{Q_\ell}. \tag{7}$$

A detailed explanation and an analysis of the implications of these approximations is given later.

With these approximations, the multicomponent equation employed by the sectional technique becomes

$$\begin{aligned}\frac{dQ_{\ell,k}}{dt} = \frac{1}{2}\sum_{i=1}^{\ell-1}\sum_{j=1}^{\ell-1}[\bar{\beta}_{i,j,\ell}^{1a}Q_{j,k}Q_i + \bar{\beta}_{i,j,\ell}^{1b}Q_{i,k}Q_j] \ - \sum_{i=1}^{\ell-1}[\bar{\beta}_{i,\ell}^{2a}Q_{\ell,k}Q_i - \bar{\beta}_{i,\ell}^{2b}Q_{i,k}Q_\ell] \\ - \frac{1}{2}\bar{\beta}_{\ell,\ell}^{3}Q_{\ell,k}Q_\ell - Q_{\ell,k}\sum_{i=\ell+1}^{m}\bar{\beta}_{i,\ell}^{4}Q_i.\end{aligned} \tag{8}$$

Because of the approximations in the sectional method, all factors containing a time and species dependence have been removed from the integrals, and hence one has:

$$\bar{\beta}_{i,j,\ell}^{1a} = \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\theta(v_{\ell-1} < u + v < v_\ell)\ u\ \beta(u,v)}{u\ v\ (x_i - x_{i-1})(x_j - x_{j-1})} dydx$$

$$\bar{\beta}^{1b}_{i,j,\ell} = \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\theta(v_{\ell-1} < u+v < v_\ell)\; v\; \beta(u,v)}{u\; v\; (x_i - x_{i-1})\big(x_j - x_{j-1}\big)} dydx$$

$$\bar{\beta}^{2a}_{i,\ell} = \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\theta(u+v > v_\ell)\; u\; \beta(u,v)}{u\; v\; (x_i - x_{i-1})(x_\ell - x_{\ell-1})} dydx \tag{9}$$

$$\bar{\beta}^{2b}_{i,\ell} = \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\theta(u+v < v_\ell)\; v\; \beta(u,v)}{u\; v\; (x_i - x_{i-1})(x_\ell - x_{\ell-1})} dydx$$

$$\bar{\beta}^{3}_{\ell,\ell} = \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\theta(u+v > v_\ell)\; (u+v)\; \beta(u,v)}{u\; v\; (x_\ell - x_{\ell-1})(x_\ell - x_{\ell-1})} dydx$$

$$\bar{\beta}^{4}_{i,\ell} = \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{u\; \beta(u,v)}{u\; v\; (x_i - x_{i-1})(x_\ell - x_{\ell-1})} dydx$$

which are independent of both time and species and are functions of section only. This is critical since these beta values can be calculated once per size section and then inserted into the differential equation as constant coefficients in time. It is this aspect that makes the sectional technique computationally feasible.

II.B. Description of DSMC Technique

In the papers by Palaniswaamy and Loyalka[6-8], they construct a Direct Simulation Monte Carlo program based upon the work by Bird[19] on rarefied gas dynamics which simulates the interaction of aerosol particles with such factors as coagulation, condensation, deposition, and source. Palsmeier and Loyalka[9] continued this work by incorporating charge into the modeling of aerosol particles. The resulting program makes it possible to explore the effects of charge on coagulation and electrostatic dispersion. Using agglomeration algorithms, this version of DSMC models the charge effects on the evolution of spatially homogeneous aerosols using a no time counter (NTC) sampling method to select particles in a list for coagulation. A brief explanation of the process involved in these DSMC programs is included which forms the foundation of this paper's Monte Carlo simulations.

In the first step, a list of particles is sampled from the desired initial distribution using a rejection-acceptance technique with each particle of the form $[mass_1 \;\; mass_2 \dots mass_s \;\; charge]$ where the first s entries are the proportions of the particle mass of that component. A multiplicative scale factor, f, is used to denote the ratio of sampled to actual particles in the simulation. Next, a time step is chosen and the NTC method developed by Bird[19] is used to determine the number of particles that will interact in that time. If selected, two particles are combined by directly adding their mass (and, if applicable, charge) and creating a new particle entry in the list with the other two entries being deleted. In this way, mass is conserved. Finally, the list is sampled for electrostatic dispersion. This process is repeated until the desired time frame has elapsed.

The great strength of this program is in its versatility. It allows for the inclusion of charge in modeling aerosol dynamics as well as more realistic multicomponent analysis. However, the DSMC technique can be computationally expensive for complex simulations. It is this program that forms the foundation for the DSMC portion of this project's computation. The scope of this project is not to study charge, so the portion of the code incorporating charge and electrostatic dispersion is removed. Other modifications were made to increase computational speed and efficiency and are described in a later section.

III. Benchmark Using Exact Solution: Single Component Aerosol

In order to verify the accuracy of the adapted DSMC program, it was decided to benchmark both it and the sectional program against an exact solution. The general dynamic equation for a coagulating aerosol is a complex, non-linear problem for which the only means of solution is a numeric technique such as the sectional method or a probabilistic technique such as DSMC. However, by making simplifications to the coagulation coefficient, it is possible to construct an exact solution for a single component aerosol. Williams and Loyalka[3] describe an exact solution for a *single component* aerosol with a "sum of masses" coagulation coefficient, $\beta(u,v) = b(u+v)$ for some constant b as first demonstrated by Scott.[20] This solution is described below and its results are compared to those of the DSMC and sectional programs.

III.A. Description of Exact Solution

The non-sectional dynamic equation for a single component aerosol distribution $n(v,t)$ under the influence of coagulation alone is given by

$$\frac{\partial}{\partial t} n(v,t) = \frac{1}{2}\int_0^v du\, \beta(u, v-u) n(u,t) n(v-u,t) - n(v,t)\int_0^\infty du\, \beta(u,v) n(u,t). \quad (10)$$

Consider the simple case when the coagulation coefficient, β, is given by $\beta(u,v) = b(u+v)$ for some constant b.

The exact solution to this equation can be computed[3,20] for an initial condition of $n(v,0) = \frac{N_0}{v_0} Exp\left(-\frac{v}{v_0}\right)$ to be:

$$n(v,t) = N_0 Exp(-b\phi t) Exp\left(-\frac{2v}{v_0}\right)\frac{1}{v}\, I_1\left(\frac{2v}{v_0}\right). \quad (11)$$

Where I_1 is the modified Bessel function of the first kind and $\phi = \int_0^\infty dv\, v\, n(v,t)$. For this benchmark case, we have chosen the following

$$N_0 = 10^6 \frac{particles}{m^3}$$

$$b = 2$$

$$v_0 = 10^{-17}$$

$$v_{min} = 0.001\ \mu m$$

$$v_{max} = 20\ \mu m$$

A plot of the initial mass distribution, $v\, n(v,0)$, for each of forty diameter sections $(m = 40)$ is given in figure 1. This is the initial distribution as perceived by the sectional method. Within the sectional technique, a choice of $x = f(v)$ has to be made. We chose the value of

$$f(v) = \log_{10}\left[\left(\frac{6v}{\pi\rho}\right)^{\frac{1}{3}} /1\mu m\right] \tag{12}$$

as used by the MAEROS User Manual[17] test problem.

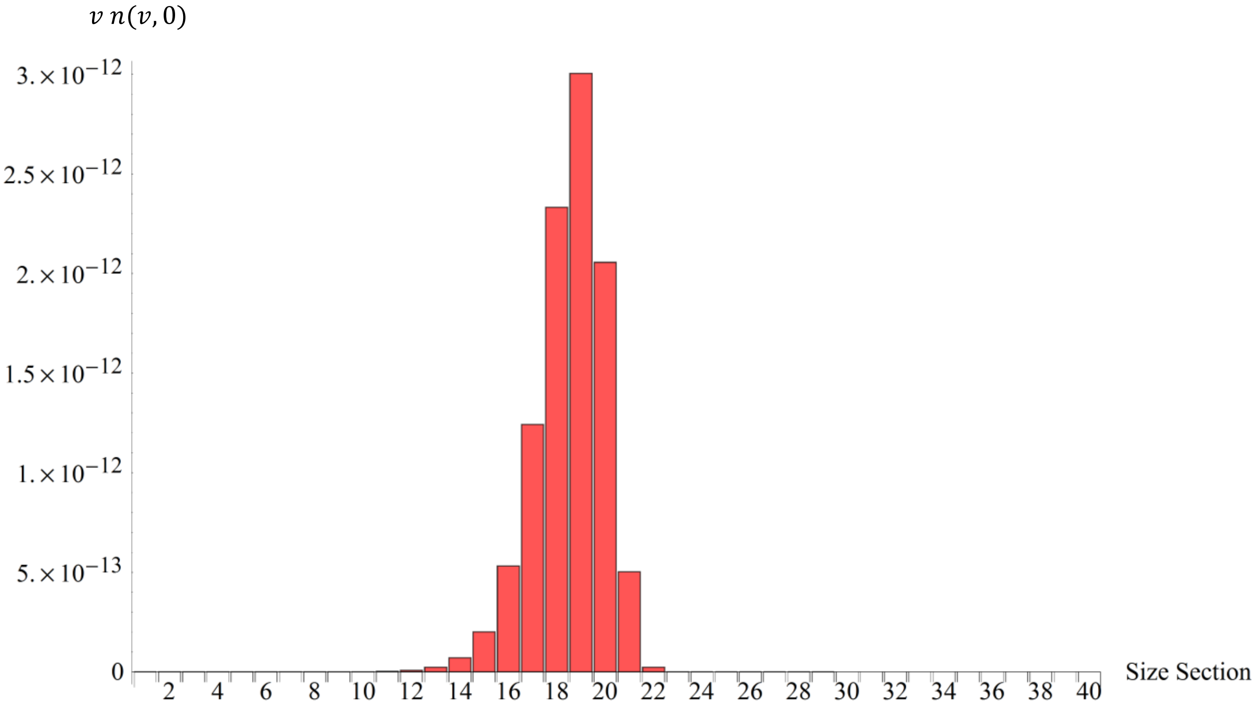


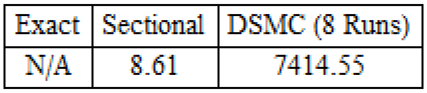
Figure 1 Initial mass distribution for the benchmark simulation over forty diameter sections.

Table I Initial parameters of the 40 diameter size sections and the initial mass density of each section for the benchmark simulation.

III.B. Comparison of Results

A graphical comparison of the DSMC and sectional programs to the exact solution at selected times is given in figures 2 through 4. Note that there are a large number of size sections in this simulation. The sectional technique is such that the accuracy increases as the number of bins increases. For this reason, we ran the simulation with a high number of bins (forty) for a more accurate comparison to the exact solution. For DSMC, eight individual runs were performed and the results were averaged with error bars showing the standard deviation from the mean. Table I gives the time taken for each of the programs to complete their computations.

Table I Computational time of the sectional and DSMC programs in seconds for the sum of masses exact solution. The time given for the DSMC simulation was for eight successive runs which were acquired for statistical purposes.



Exact	Sectional	DSMC (8 Runs)
N/A	8.61	7414.55

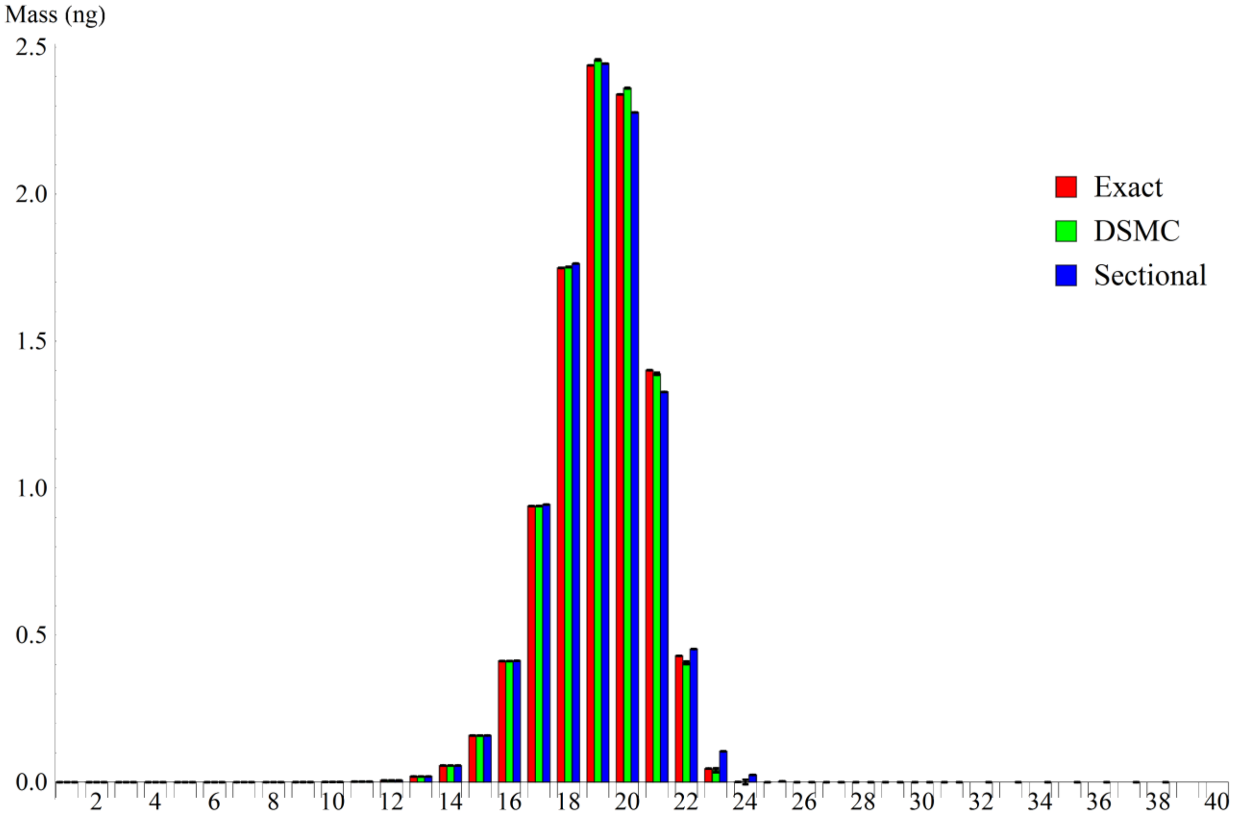


Figure 2 Comparison of DSMC and Sectional mass distributions to the sum of masses exact solution after $t = 1.0 \times 10^{10}$s.

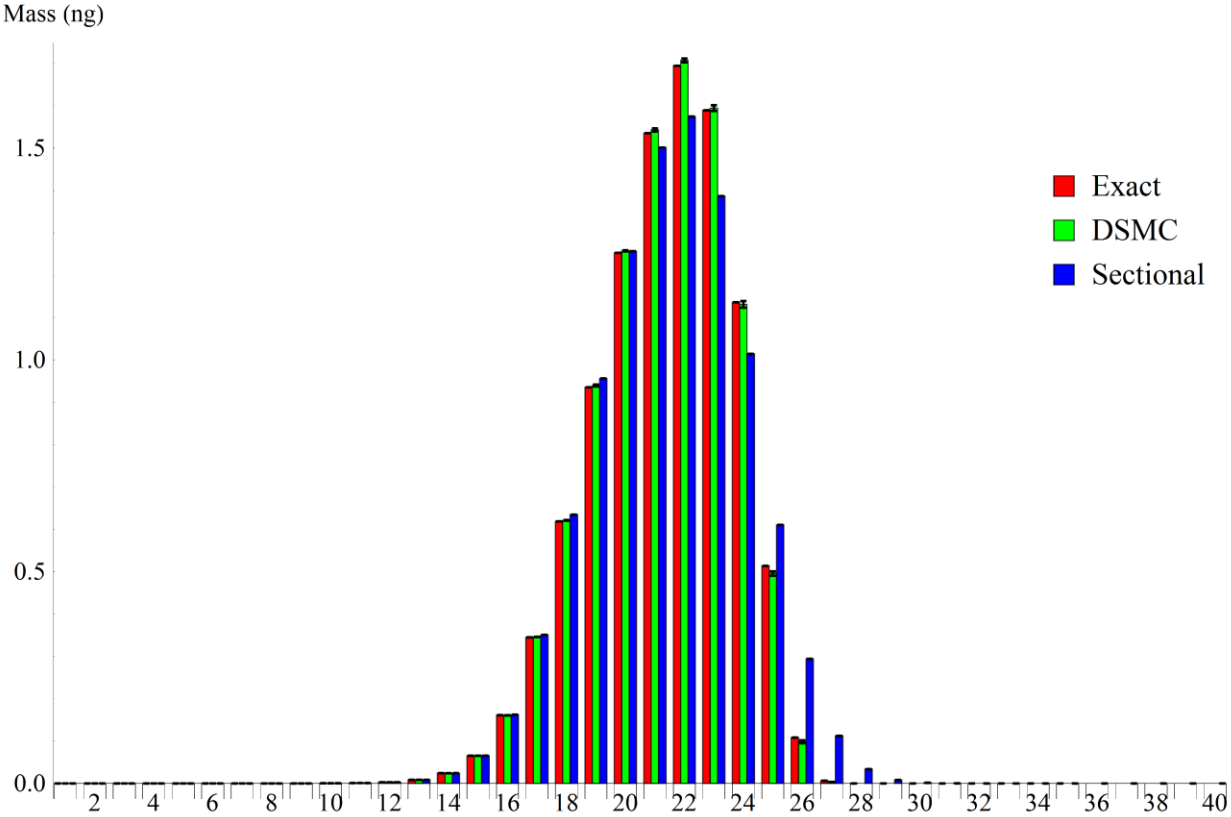


Figure 3 Comparison of DSMC and Sectional mass distributions to the sum of masses exact solution after t=5.0×10^10s.

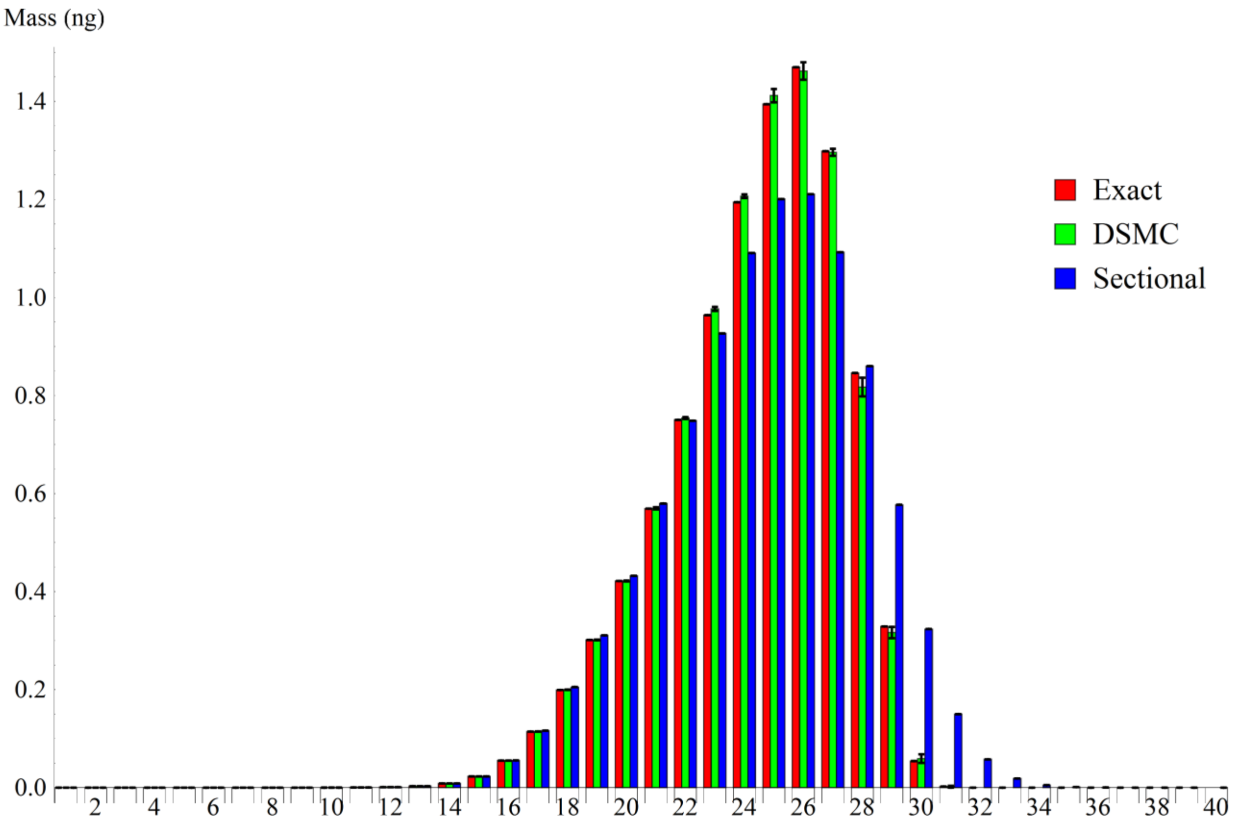


Figure 4 Comparison of DSMC and Sectional mass distributions to the sum of masses exact solution after t=1.0×10^11s.

As can be seen from the plots, both programs faithfully modeled the coagulation of the aerosol for the sum of masses problem. What difference does exist in the solutions can be easily explained when understanding the nature of each program. For the sectional method, the major deviation from the exact solution occurs in higher sections. These sections are much larger and hence the averaging on the number distribution is more significant. As for the DSMC method, deviation occurred in those bins with very few particles since the particle statistics were low. This could be remedied by averaging over additional runs or by modeling more particles. For both of these simulations, then, the limitation to accuracy is computational expense.

IV. Assumptions in Modeling of Aerosols: Intrasectional Number Distributions

As stated before, Gelbard and Seinfeld[16] make two assumptions to close the general sectional equation for aerosol coagulation. The first assumption they make is on the number distribution of the aerosol within each section. By definition, $Q_\ell(t) = \int_{v_{\ell-1}}^{v_\ell} v\, n(v,t)dv$ is the total mass within section ℓ, that is, the interval from $v_{\ell-1} < v < v_\ell$. Given a user-defined size variable of interest, x, which is uniquely related to particle mass by $x = f(v)$, the function $\bar{q}_\ell(t)$ is defined by the relationship

$$v\, n(v,t) = \bar{q}_\ell(t) f'(v), \qquad v_{\ell-1} < v < v_\ell. \tag{13}$$

Inserting this into the definition of Q_ℓ, we see that

$$\begin{aligned} Q_\ell(t) &= \int_{v_{\ell-1}}^{v_\ell} v\, n(v,t)dv \\ &= \bar{q}_\ell(t)[f(v_\ell) - f(v_{\ell-1})] \\ &= \bar{q}_\ell(t)[x_\ell - x_{\ell-1}] \end{aligned} \tag{14}$$

From the definition, we can see that the formula for the number density distribution at a given time is given by

$$n(v,t) = \bar{q}_\ell(t)\frac{f'(v)}{v} = \frac{Q_\ell(t)\, f'(v)}{v\,(f(v_\ell) - f(v_{\ell-1}))}, \qquad v_{\ell-1} < v < v_\ell. \tag{15}$$

This expression then serves as a relationship between the functions $n(v,t)$ and $Q_\ell(t)$.

There is a great deal of flexibility to the user to define the value of $x = f(v)$ to be a size variable for which the distribution is constant over each section. The choice in this variable is usually dependent upon what format the user would prefer when plotting the simulation results. The function $\bar{q}_\ell(t)$ on intervals $(x_{\ell-1}, x_\ell)$ will plot like a series of step function with $\ell = 1, \dots, m$. The area under each step will be equal to $Q_\ell(t)$ since

$$\bar{q}_\ell(t) = \frac{Q_\ell}{[x_\ell - x_{\ell-1}]}. \tag{16}$$

For atmospheric aerosols, it is common to define x by the function

$$f(v) = \log_{10}[D(v)] \tag{17}$$

for $D(v) = \left(\frac{6v}{\pi\rho}\right)^{\frac{1}{3}} / 1\mu m$ so that a plot of $\bar{q}_\ell(t)$ versus x will be a series of step functions with respect to the log of the particle diameter in microns. For a simpler expression, one can also choose

$$f(v) = v \tag{18}$$

so that x is simply particle mass. If the desire is to display the number distribution as constant versus particle mass over each section, one can choose a function of the form

$$f(v) = v^2 + C \tag{19}$$

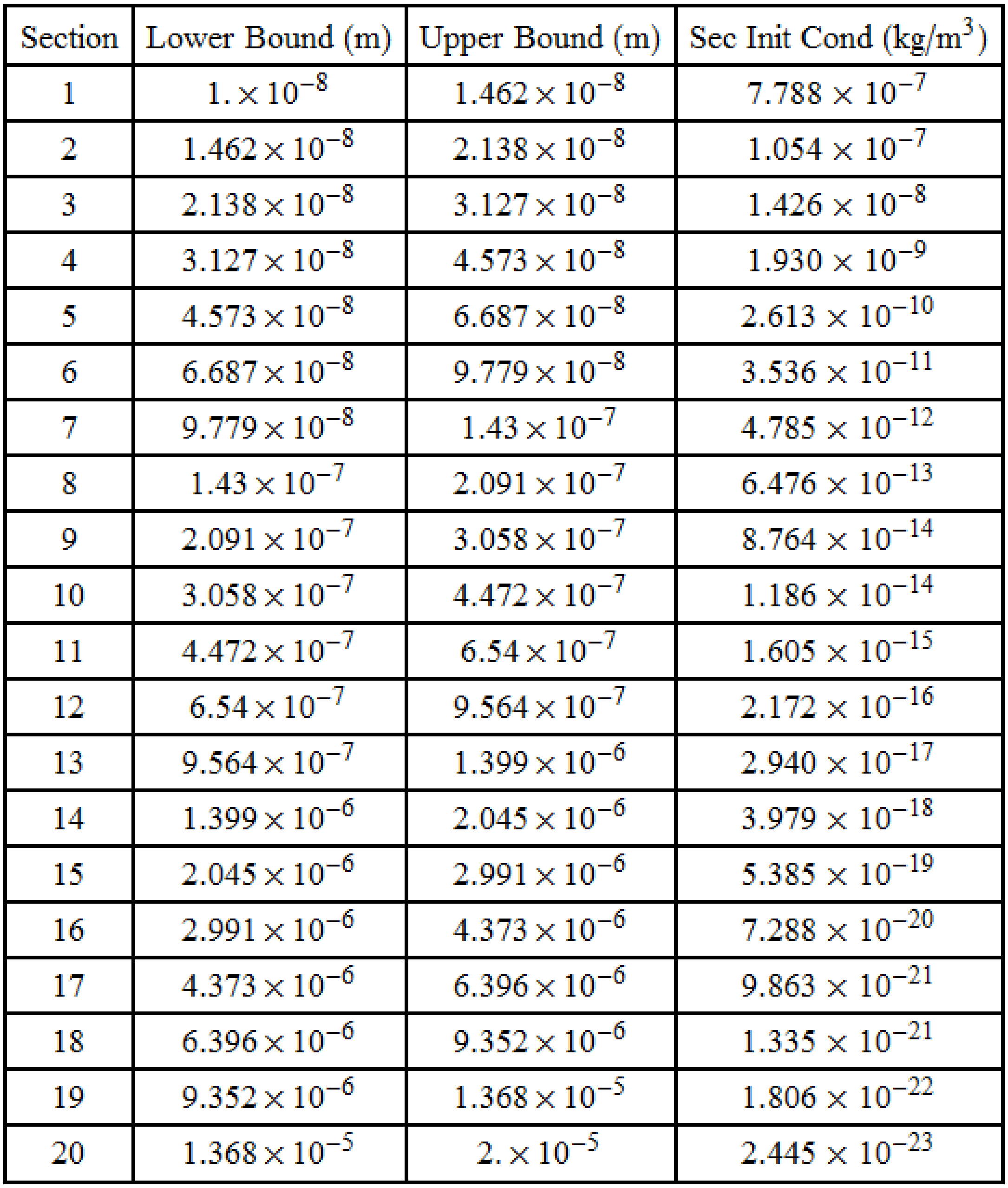
where C is some constant.

What is not discussed by Gelbard and Seinfeld, however, is what effect this choice in $x = f(v)$ has on the accuracy of the simulation. Given the parameters of the sample problem from the MAEROS User Manual[17] a comparison was made of $Q_\ell(t)$ for the sectional technique when using the three aforementioned choices of $x = f(v)$. The initial aerosol is composed of a single component and the mass concentration in each section ℓ is given by

$$Q_\ell(t = 0) = 10^{-6} Exp\left[-\frac{8(\ell - 1) + 1}{4}\right]\left(\frac{kg}{m^3}\right). \tag{20}$$

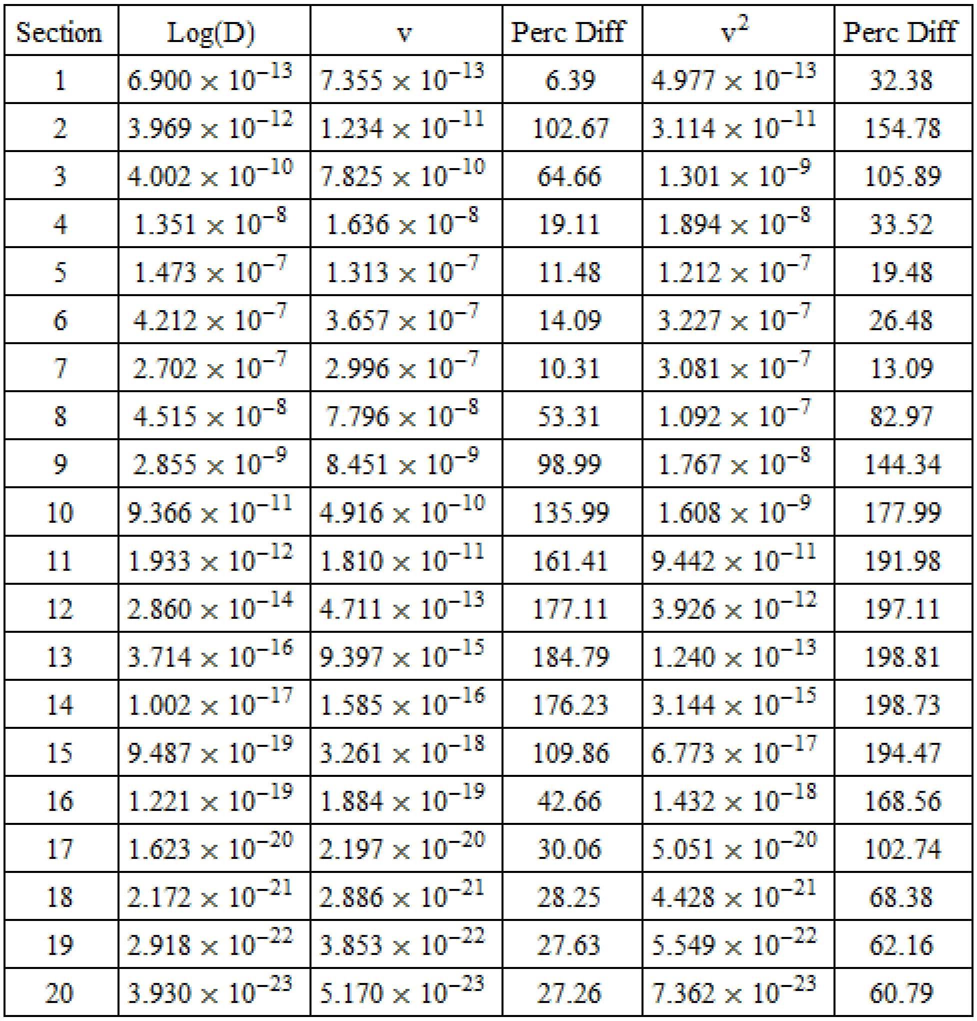
The size sections (with $v_{min} = 0.001\ \mu m$, $v_{max} = 20\ \mu m$ and $m = 20$) as well as the initial mass concentration in each is given in table II. A comparison of the mass concentration at time $t = 200.0$s is given in table III. Also included is the percent difference of $f(v) = v$ and $f(v) = v^2$ from what is assumed to be the base case of $f(v) = Log(D(v))$ used by the MAEROS user manual test problem.[17]

Table II Initial conditions as described by the MAEROS user manual[17]. The upper and lower bounds of the section indicate the upper and lower limit of the diameter of the particles within that size section. The right column gives the mass concentration $Q_{\ell,k}$ fed as initial conditions to each of the three simulations of the sectional technique.



Section	Lower Bound (m)	Upper Bound (m)	Sec Init Cond (kg/m^3)
1	$1. \times 10^{-8}$	1.462×10^{-8}	7.788×10^{-7}
2	1.462×10^{-8}	2.138×10^{-8}	1.054×10^{-7}
3	2.138×10^{-8}	3.127×10^{-8}	1.426×10^{-8}
4	3.127×10^{-8}	4.573×10^{-8}	1.930×10^{-9}
5	4.573×10^{-8}	6.687×10^{-8}	2.613×10^{-10}
6	6.687×10^{-8}	9.779×10^{-8}	3.536×10^{-11}
7	9.779×10^{-8}	1.43×10^{-7}	4.785×10^{-12}
8	1.43×10^{-7}	2.091×10^{-7}	6.476×10^{-13}
9	2.091×10^{-7}	3.058×10^{-7}	8.764×10^{-14}
10	3.058×10^{-7}	4.472×10^{-7}	1.186×10^{-14}
11	4.472×10^{-7}	6.54×10^{-7}	1.605×10^{-15}
12	6.54×10^{-7}	9.564×10^{-7}	2.172×10^{-16}
13	9.564×10^{-7}	1.399×10^{-6}	2.940×10^{-17}
14	1.399×10^{-6}	2.045×10^{-6}	3.979×10^{-18}
15	2.045×10^{-6}	2.991×10^{-6}	5.385×10^{-19}
16	2.991×10^{-6}	4.373×10^{-6}	7.288×10^{-20}
17	4.373×10^{-6}	6.396×10^{-6}	9.863×10^{-21}
18	6.396×10^{-6}	9.352×10^{-6}	1.335×10^{-21}
19	9.352×10^{-6}	1.368×10^{-5}	1.806×10^{-22}
20	1.368×10^{-5}	$2. \times 10^{-5}$	2.445×10^{-23}

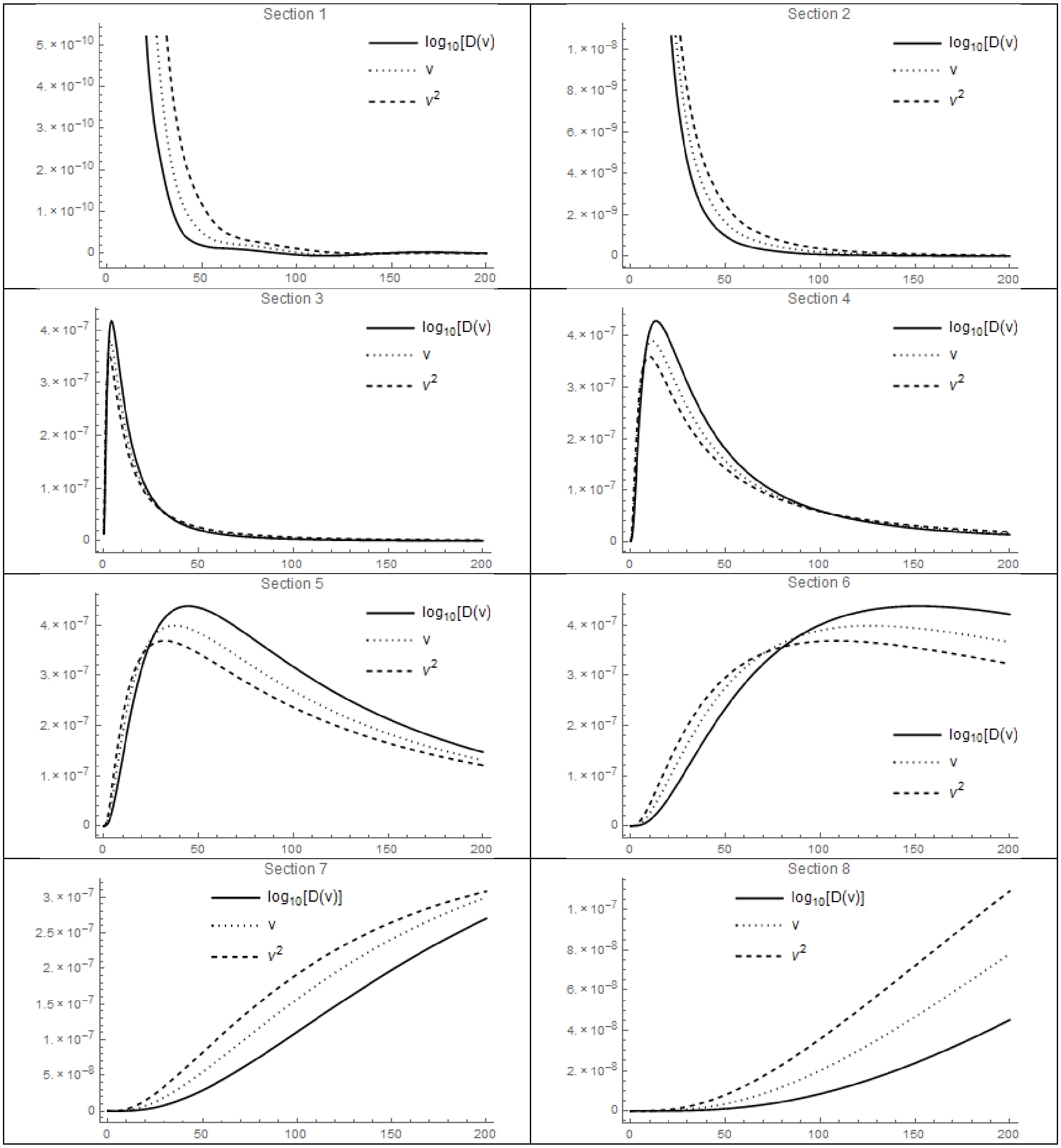
Table III Mass concentration (kg/m^3) in each section after $t = 200s$ of simulation time for three different values of $f(v)$. The "Perc Diff" represents the percent difference of the given value from the corresponding one from the base case of the Log(D(v)) simulation.

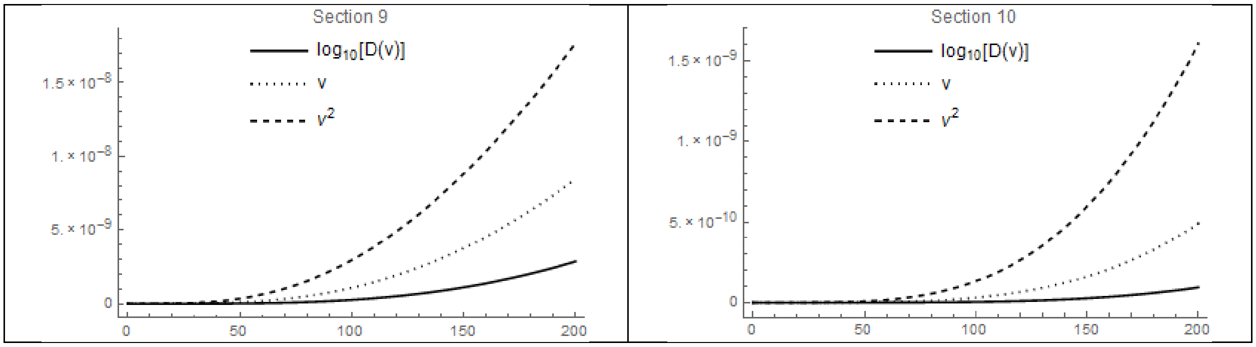


Section	Log(D)	v	Perc Diff	v^2	Perc Diff
1	6.900×10^{-13}	7.355×10^{-13}	6.39	4.977×10^{-13}	32.38
2	3.969×10^{-12}	1.234×10^{-11}	102.67	3.114×10^{-11}	154.78
3	4.002×10^{-10}	7.825×10^{-10}	64.66	1.301×10^{-9}	105.89
4	1.351×10^{-8}	1.636×10^{-8}	19.11	1.894×10^{-8}	33.52
5	1.473×10^{-7}	1.313×10^{-7}	11.48	1.212×10^{-7}	19.48
6	4.212×10^{-7}	3.657×10^{-7}	14.09	3.227×10^{-7}	26.48
7	2.702×10^{-7}	2.996×10^{-7}	10.31	3.081×10^{-7}	13.09
8	4.515×10^{-8}	7.796×10^{-8}	53.31	1.092×10^{-7}	82.97
9	2.855×10^{-9}	8.451×10^{-9}	98.99	1.767×10^{-8}	144.34
10	9.366×10^{-11}	4.916×10^{-10}	135.99	1.608×10^{-9}	177.99
11	1.933×10^{-12}	1.810×10^{-11}	161.41	9.442×10^{-11}	191.98
12	2.860×10^{-14}	4.711×10^{-13}	177.11	3.926×10^{-12}	197.11
13	3.714×10^{-16}	9.397×10^{-15}	184.79	1.240×10^{-13}	198.81
14	1.002×10^{-17}	1.585×10^{-16}	176.23	3.144×10^{-15}	198.73
15	9.487×10^{-19}	3.261×10^{-18}	109.86	6.773×10^{-17}	194.47
16	1.221×10^{-19}	1.884×10^{-19}	42.66	1.432×10^{-18}	168.56
17	1.623×10^{-20}	2.197×10^{-20}	30.06	5.051×10^{-20}	102.74
18	2.172×10^{-21}	2.886×10^{-21}	28.25	4.428×10^{-21}	68.38
19	2.918×10^{-22}	3.853×10^{-22}	27.63	5.549×10^{-22}	62.16
20	3.930×10^{-23}	5.170×10^{-23}	27.26	7.362×10^{-23}	60.79

Note the vast difference between the results of the sectional technique depending upon the choice of the variable $x = f(v)$. Clearly, the choice in this value has a much greater impact on the simulation than simply determining the type of plot for the results. In the smaller sections, as the simulation evolves in time and the mass density is decreasing and moving into the larger sections, the three distributions do not diverge as severely as the larger sections and, hence, for longer simulations, the choice of x has less (though still significant) of an impact on these sections. However, in a short period of time the difference in each becomes substantial with it differing as much as an order of magnitude. The figures in table IV further demonstrate this by comparing the time evolution of the mass density in the first ten sections.

Table IV Plots of the time evolution (in seconds) of the mass concentration (in kg/m^3) for the first ten size sections. These plots demonstrate the impact the choice in x = f(v) has on the results of a simulation.





This simulation has demonstrated that the choice of $x = f(v)$ has a significant impact on the results of the simulation and that the user should be critical of their choice and the impact it may have on the type of simulation being performed. However, it fails to indicate *which* value should be employed to achieve the greatest fidelity to the aerosol physics. We seek to demonstrate whether the user, knowing their initial aerosol distribution, could also input a value for x that would be most favorable to their simulation. Given the exact solution described in the previous section with an approximate sum of mass coagulation kernel, a comparison was made between the number distribution, $n(v,0) = \frac{N_0}{v_0} Exp\left(-\frac{v}{v_0}\right)$, for the exact solution and that of the sectional technique, $n(v,0) = \frac{Q_\ell(0)\, f\prime(v)}{v\,(f(v_\ell)-f(v_{\ell-1}))}$, when employing the three aforementioned choices of $x = f(v)$. The initial number distributions resulting from these three functions are given in the following figures for a sample of nine sections.

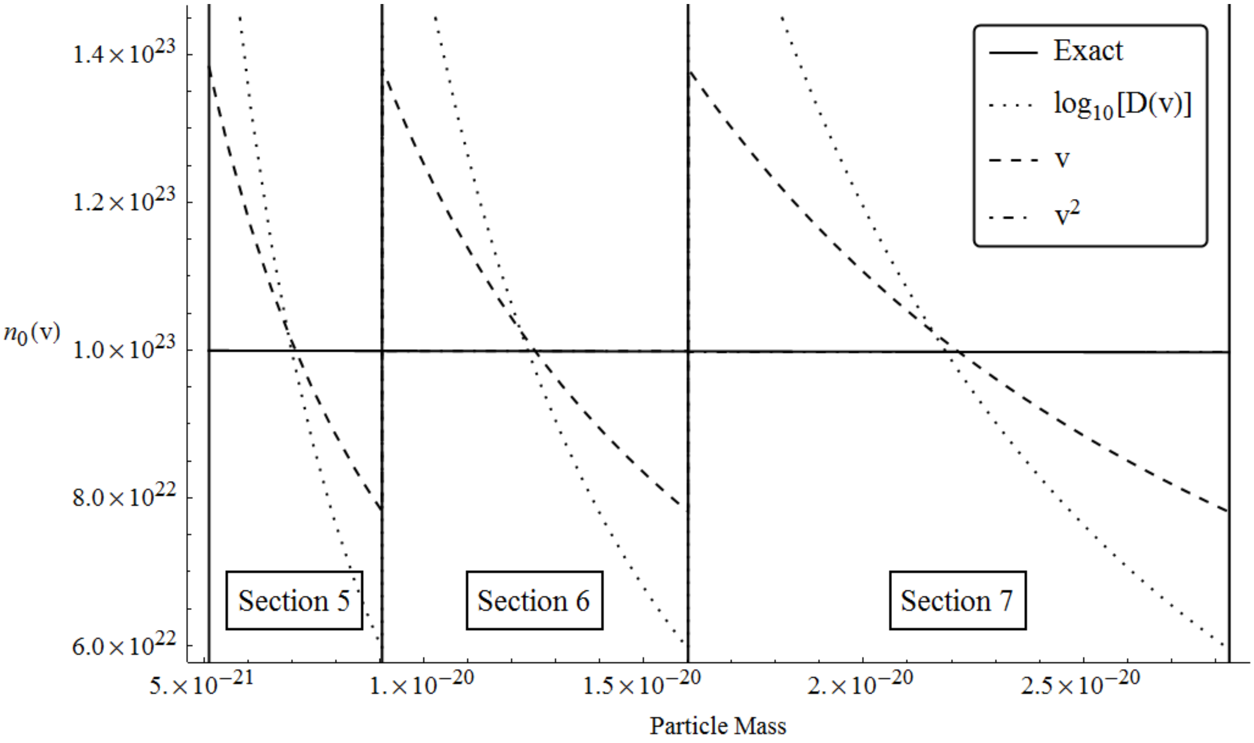


Figure 5 A plot of the number distribution $n(v,0)$ resulting from the various values of $f(v)$ in the sectional technique.

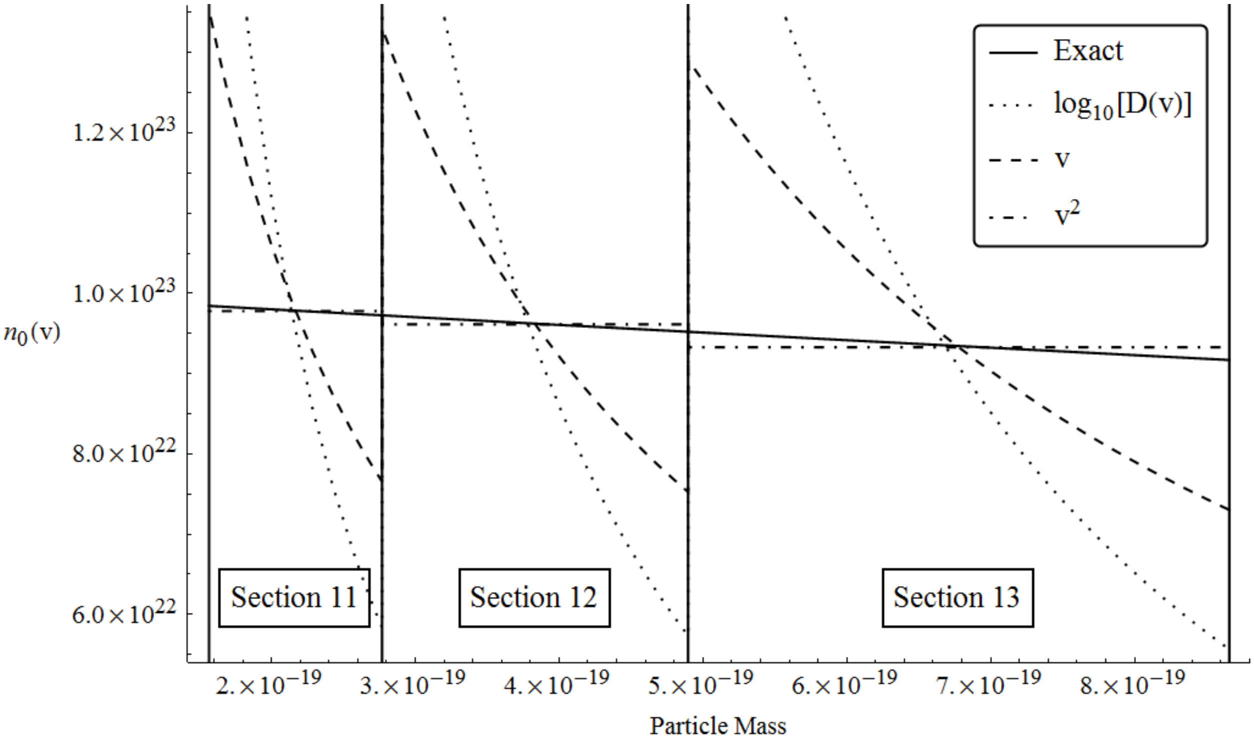


Figure 6 A plot of the number distribution $n(v,0)$ resulting from the various values of $f(v)$ in the sectional technique.

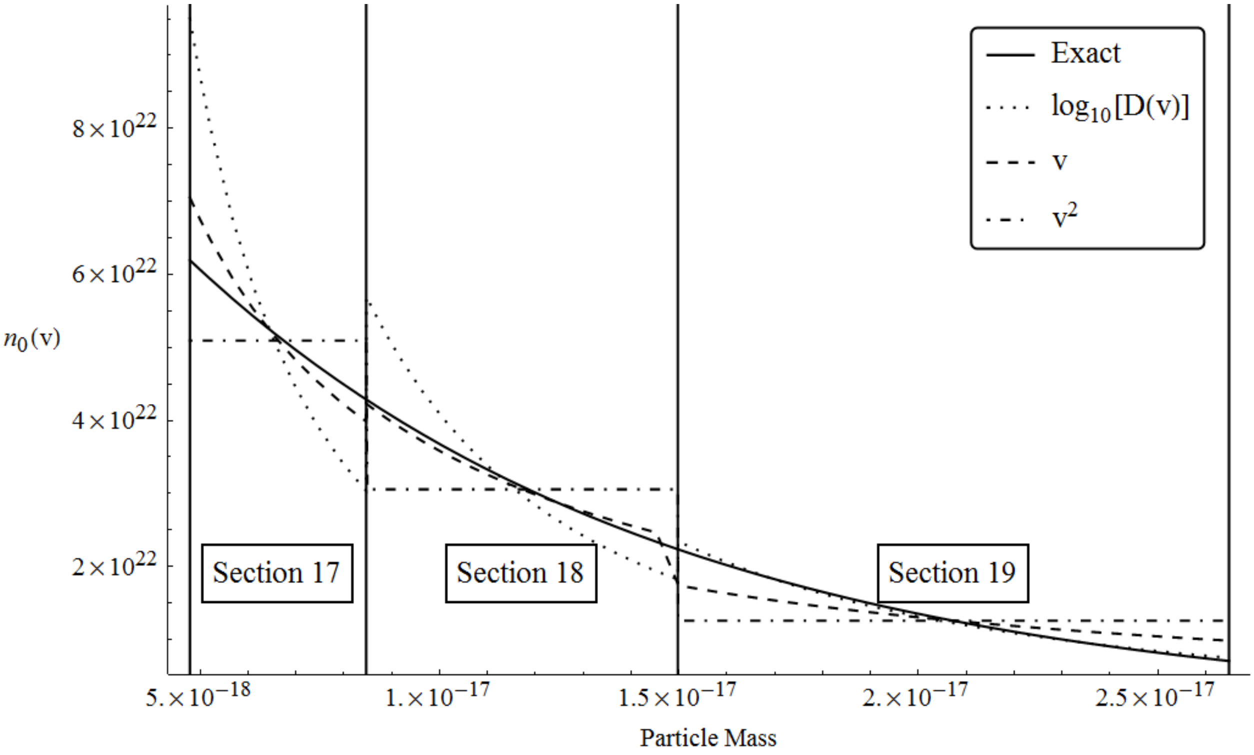


Figure 7 A plot of the number distribution $n(v, 0)$ resulting from the various values of $f(v)$ in the sectional technique.

Notice that in the smaller sections, the number distribution that most accurately aligns with the exact distribution is the one corresponding to $f(v) = v^2$. In these sections, the width of the section is so small that it is practically linear on this interval even though in reality the exact number distribution is actually quite steep (an exponential function). However, the exponential nature of the distribution becomes more apparent as the width of the intervals become larger. For this reason, the other distributions, in particular the one corresponding to $f(v) = \log(D(v))$, becomes increasingly more appealing in the larger sections as seen in figure 7. This difference in the number distribution in each section has a direct impact on the accuracy of the simulation. This is demonstrated by table V which compares the mass density in each section for the three sectional simulations to the exact solution at simulation time $t = 1.\times 10^{11}$s. Notice that in the larger sections the percent error from the exact $\left(P.E. = 100\% \cdot \frac{|x-exact|}{exact}\right)$ is diminished in the logarithmic simulation as compared to the other simulations revealing a direct correlation between the accuracy of the number distribution due to choice of x and the accuracy of the simulation.

While we have explored the choice of an optimal $f(v)$ for comparisons with an exact solution in time, it does not follow that our conclusions will apply to cases with realistic coagulation kernels. This is a topic we wish to explore in a future paper by comparing sectional simulations with various $f(v)$ to DSMC simulations. It is safe to conclude, however, that careful consideration should be given to the choice of this size variable and how closely its corresponding number distribution will compare to the actual distribution, particularly in the larger size sections. Also, we have seen that the optimal choice of x changes from section to section. The best choice of x not only depends upon the shape of the aerosol number distribution, but also upon the time under consideration within the simulation.

Table V Mass in kilograms in each size section at time $t = 1.\times 10^{11}$s for the exact solution, sectional with $f(v) = log(D(v))$, sectional with $f(v) = v$, and sectional with $f(v) = v^2$.

Section	Exact	Log(D)	Perc Error	v	Perc Error	v^2	Perc Error
1	3.94571×10^{-21}	3.926×10^{-21}	0.49	3.929×10^{-21}	0.43	3.930×10^{-21}	0.39
2	1.23377×10^{-20}	1.228×10^{-20}	0.49	1.228×10^{-20}	0.43	1.229×10^{-20}	0.39
3	3.8575×10^{-20}	3.839×10^{-20}	0.49	3.841×10^{-20}	0.43	3.843×10^{-20}	0.39
4	1.20591×10^{-19}	1.200×10^{-19}	0.49	1.201×10^{-19}	0.43	1.201×10^{-19}	0.38
5	3.7689×10^{-19}	3.750×10^{-19}	0.49	3.753×10^{-19}	0.43	3.754×10^{-19}	0.38
6	1.17738×10^{-18}	1.172×10^{-18}	0.49	1.172×10^{-18}	0.43	1.173×10^{-18}	0.38
7	3.67512×10^{-18}	3.657×10^{-18}	0.49	3.659×10^{-18}	0.43	3.661×10^{-18}	0.38
8	1.14555×10^{-17}	1.140×10^{-17}	0.49	1.141×10^{-17}	0.42	1.141×10^{-17}	0.37
9	3.56184×10^{-17}	3.545×10^{-17}	0.48	3.547×10^{-17}	0.42	3.549×10^{-17}	0.36
10	1.10262×10^{-16}	1.097×10^{-16}	0.48	1.098×10^{-16}	0.4	1.099×10^{-16}	0.34
11	3.38711×10^{-16}	3.371×10^{-16}	0.47	3.374×10^{-16}	0.38	3.377×10^{-16}	0.31
12	1.02653×10^{-15}	1.022×10^{-15}	0.45	1.023×10^{-15}	0.34	1.024×10^{-15}	0.25
13	3.03912×10^{-15}	3.027×10^{-15}	0.39	3.031×10^{-15}	0.27	3.034×10^{-15}	0.16
14	8.6452×10^{-15}	8.622×10^{-15}	0.27	8.633×10^{-15}	0.14	8.645×10^{-15}	0.
15	2.30158×10^{-14}	2.302×10^{-14}	0.03	2.304×10^{-14}	0.1	2.307×10^{-14}	0.21
16	5.52164×10^{-14}	5.559×10^{-14}	0.67	5.547×10^{-14}	0.47	5.545×10^{-14}	0.42
17	1.14308×10^{-13}	1.163×10^{-13}	1.78	1.152×10^{-13}	0.81	1.146×10^{-13}	0.24
18	1.99331×10^{-13}	2.052×10^{-13}	2.94	2.002×10^{-13}	0.45	1.968×10^{-13}	1.27
19	3.01416×10^{-13}	3.108×10^{-13}	3.13	2.970×10^{-13}	1.47	2.866×10^{-13}	4.93
20	4.21746×10^{-13}	4.323×10^{-13}	2.51	4.028×10^{-13}	4.49	3.797×10^{-13}	9.96
21	5.69281×10^{-13}	5.796×10^{-13}	1.81	5.256×10^{-13}	7.68	4.828×10^{-13}	15.2
22	7.50249×10^{-13}	7.486×10^{-13}	0.22	6.637×10^{-13}	11.54	5.955×10^{-13}	20.62
23	9.63968×10^{-13}	9.267×10^{-13}	3.86	8.091×10^{-13}	16.06	7.129×10^{-13}	26.05
24	1.19436×10^{-12}	1.091×10^{-12}	8.67	9.475×10^{-13}	20.67	8.260×10^{-13}	30.84
25	1.39429×10^{-12}	1.201×10^{-12}	13.88	1.054×10^{-12}	24.42	9.205×10^{-13}	33.98
26	1.46976×10^{-12}	1.211×10^{-12}	17.61	1.099×10^{-12}	25.23	9.777×10^{-13}	33.48
27	1.29838×10^{-12}	1.092×10^{-12}	15.87	1.059×10^{-12}	18.44	9.802×10^{-13}	24.51
28	8.46008×10^{-13}	8.600×10^{-13}	1.65	9.282×10^{-13}	9.72	9.184×10^{-13}	8.55
29	3.28935×10^{-13}	5.770×10^{-13}	75.42	7.282×10^{-13}	121.39	7.962×10^{-13}	142.04
30	5.43094×10^{-14}	3.236×10^{-13}	495.76	5.036×10^{-13}	827.32	6.324×10^{-13}	1064.48

V. Benchmark Using Exact Solution: Multicomponent Aerosol

In an effort to validate the sectional and DSMC programs' handling of a multicomponent aerosol, it was decided to compare them with a multicomponent exact solution. Williams and Loyalka[3] describe an exact solution for a *two component* aerosol with a constant coagulation coefficient, $\beta(u, v) = c$, which was first demonstrated by Lushkinov.[21] Here we compare the results of the sectional and DSMC methods for a simple simulation.

In the case of a 2 species aerosol for which the coagulation coefficient is assumed to be a constant, the exact solution for $n(v_1, v_2, t)$ can be computed. We consider the case in which $n(v_1, v_2, t = 0) = \frac{N_0}{v_{10}v_{20}} Exp\left(-\frac{v_1}{v_{10}} - \frac{v_2}{v_{20}}\right)$ for v_{10} and v_{20} the average initial diameters of the particles. The general solution in this case is,

$$n(v_1, v_2, t) = \frac{N_0}{v_{10}\, v_{20}(1+\tau)^2} Exp\left(-\frac{v_1}{v_{10}} - \frac{v_2}{v_{20}}\right)\, I_0\left(2\left(\frac{\tau\, v_1\, v_2}{(1+\tau)\, v_{10}\, v_{20}}\right)\right)^{1/2} \quad (21)$$

where I_0 is the modified Bessel function and $\tau = \frac{\beta_0 N_0\, t}{2}$. For this benchmark case, we have chosen the following

$$N_0 = 10^6 \frac{particles}{m^3}$$

$$\beta_0 = 1$$

$$v_{10} = 0.07\, \mu m$$

$$v_{20} = 0.08\, \mu m$$

$$v_{min} = 0.001\, \mu m$$

$$v_{max} = 20\ \mu m$$

The initial mass concentration (kg/m^3) for each component k is given by $M_k(v) = \int n(v_1, v_2, 0)\rho\ v_k \delta(v_1 + v_2 = v) dv_k$. The initial sectional distribution for each section ℓ is then given by $\int_{v_{\ell-1}}^{v_\ell} M_k(v) dv$. For the DSMC simulation, we generated a random sampling of the diameter distribution and gave each particle an identical component mass distribution within each size section (figure 15 demonstrates this) and averaged the results over eight simulations. The initial distribution and the final distribution at time $t = 10^{-5} s$ are given in the following figures for sections 16 through 26. In each section, the first bar represents the exact solution, the second is DSMC, and the third is for the sectional solution.

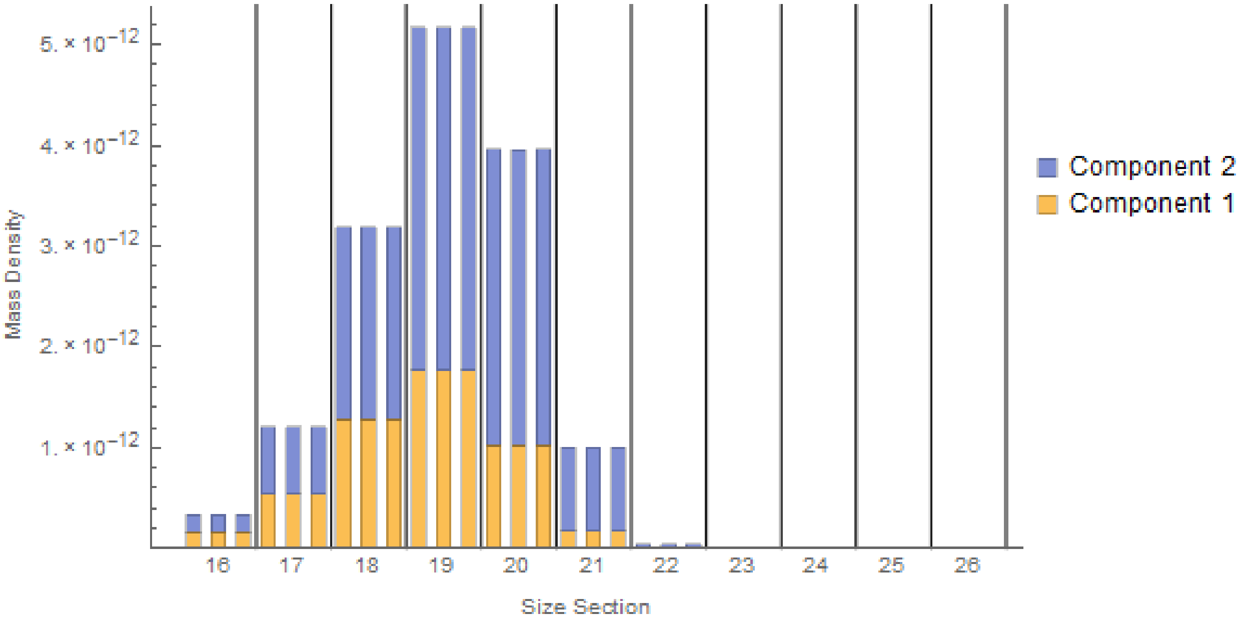


Figure 8 Initial distributions of the Exact, DSMC and Sectional simulations. In each size section, the first column represents the exact solution; the second, DSMC; and the third, sectional.

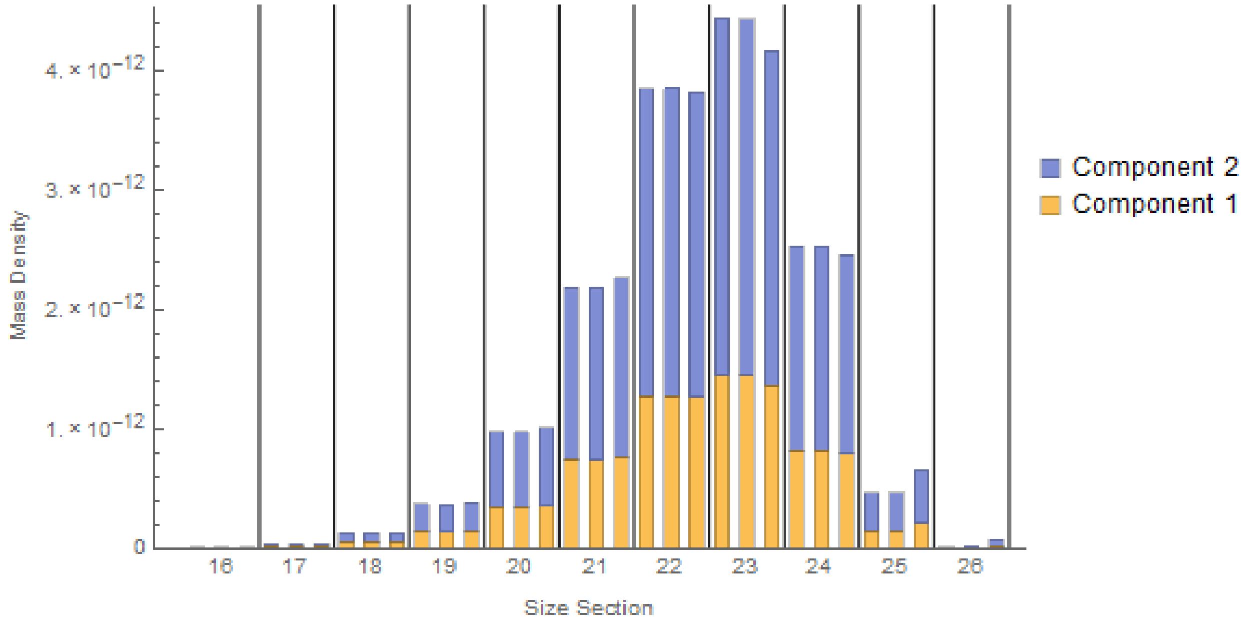


Figure 9 Comparison of the DSMC and Sectional techniques to an exact solution with a constant coagulation coefficient. In each size section, the first column represents the exact solution; the second, DSMC; and the third, sectional. Error bars were not included for DSMC since the error was negligible.

We note here that both the sectional and DSMC methods follow the multi-component exact solution in time. In particular, DSMC aligns with the exact in all sections while the sectional simulation begins to deviate in the larger sections. While assuming a constant coagulation kernel is an especially simplified case, it strengthens the idea that DSMC faithfully models the particle physics of a multicomponent aerosol.

VI. Additional Assumptions in Modeling of Multicomponent Aerosols: Intrasectional Component Distributions

The sectional method has been shown[5- 8] to be insufficient in many cases of multi-component aerosols due to assumptions made on the aerosol. As mentioned earlier, one assumption necessary for this technique is that each particle in the system have the same mass density regardless of component type. Especially when considering the coagulation kernel, $\beta(u, v)$, the effect of this assumption is readily apparent. The rate of agglomeration strongly depends upon the density of the particles involved. By assuming all components have the same density, a great deal of precision is lost in the simulation of multicomponent aerosols where differences in density can often be an order of magnitude.

A second assumption on the multicomponent aerosol is made with regards to the distribution of the components in each section in order to close the sectional equation. This assumption is more involved and its impact on the accuracy of the method is less apparent. Gelbard and Seinfeld[16] define the function $g_k(v, \eta_k)d\eta_k$ as the fraction of aerosol particles with mass in the range $[v, v + dv]$ and a mass fraction of component k in the range $[\eta_k, \eta_k + d\eta_k]$. This function acts as a probability density function for the components of the aerosol (i.e., for a given component k, g_k gives the fraction of particles of mass v with a mass fraction of component k in the range $[\eta_k, \eta_k + d\eta_k]$). The assumption made by the sectional technique regarding multicomponent aerosols is that for all particles within a section, the mean mass fraction of each component k is equal (though distinct from section to section). Using the terminology of a density function, this equates to assuming that, within a section, the expected value of the mass fraction of component k, $E[\eta_k] = \int_0^1 \eta_k g_k(v, \eta_k)d\eta_k$, is independent of v. The following figure illustrates how for a given size section, ℓ, a component distribution is averaged out by the sectional method.

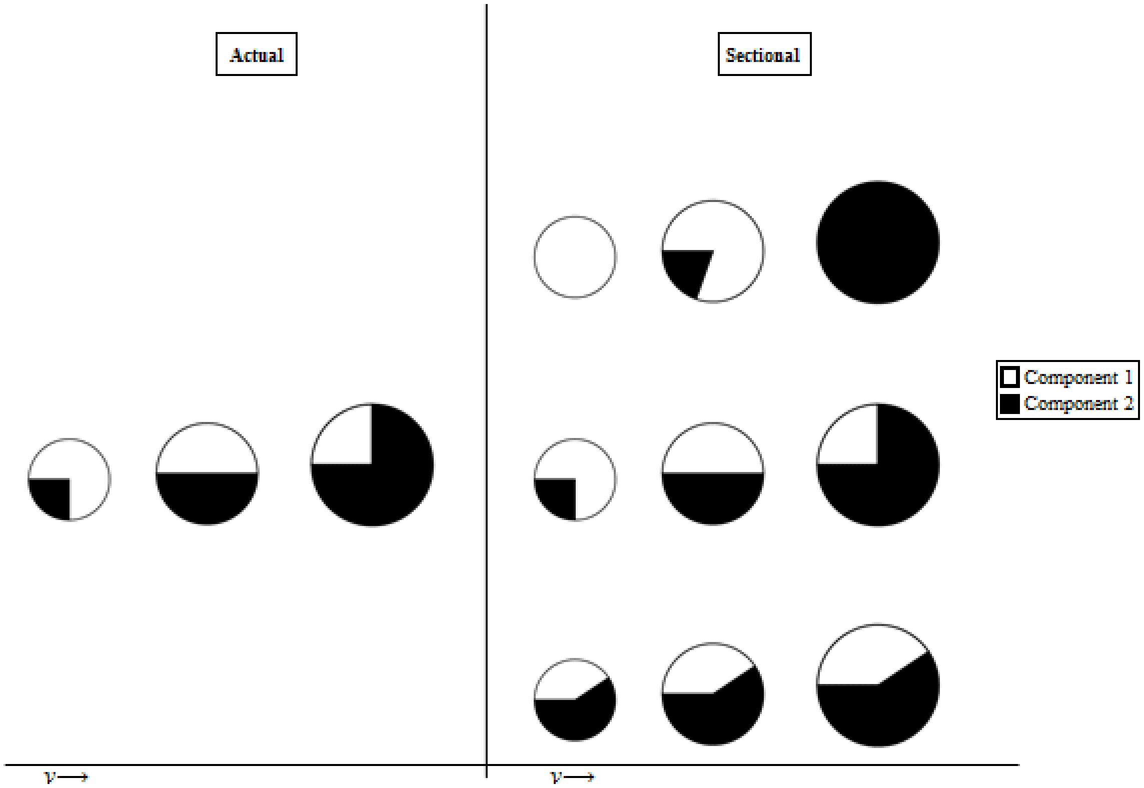


Figure 10 Demonstration of the approximation made by the sectional technique on the component distribution for a two component aerosol. For a given section, ℓ, the left hand side represents an aerosol for which $\int_0^1 \eta_k g_k(v,\eta_k) d\eta_k$ varies with v. The right hand side demonstrates three distributions which are seen as equal under the approximation by the sectional technique. The technique makes no assumption on the component distribution and sees no difference between each of the example cases shown.[16]

The function $G_k(v) = \int_0^1 \eta_k g_k(v,\eta_k) d\eta_k$ appears within the sectional equation for a multicomponent aerosol in the definition of $\bar{v}_{\ell,k}$, which is given as

$$\bar{v}_{\ell,k}(v) = v \int_0^1 \eta_k g_k(v,\eta_k) d\eta_k = v\, G_k(v). \tag{22}$$

For a given particle mass v, $\bar{v}_{\ell,k}$ represents the mean mass of component k in the range $[v, v + dv]$. It is this value, then, of $G_k(v)$ that is averaged out within each section with the implementation of this second assumption.

This is a major weakness of this technique, namely, the inability of the sectional technique to distinguish among different component distributions. To illustrate this, consider an aerosol with two components of the same density (since the sectional technique necessitates this) for which the composition of the particle is strongly dependent on its size as described in figure 11. The vertical line in this figure represents the boundary of the size section. Here we see that $G_1(5.24 \times 10^{-22}) = 1$ so a particle of mass $5.24 \times 10^{-22}\ kg$ will be composed entirely of component one. Further, a particle of mass $2.16 \times 10^{-21} kg$ has $G_1 = G_2 = .5$ so it will be made up of half components one and two. In contrast, consider a different aerosol in which the fraction of mass attributed to components one and two is assumed constant within the section as demonstrated by figure 12.

The sectional technique averages out the value of $G_k(v)$. That is, the sectional approximation assumes the amount of mass in section one for components one and two is given by

$$m_{comp1} = N \int_{v_0}^{v_1} v\ G_1(v) dv \tag{23}$$

$$m_{comp2} = N \int_{v_0}^{v_1} v\ G_2(v) dv. \tag{24}$$

where N represent the number of particles within the mass range $[v_0, v_1]$ of section one. Hence, the sectional technique will see both of the aforementioned cases as having the same value of m_{comp1} and m_{comp2} and therefore identical.

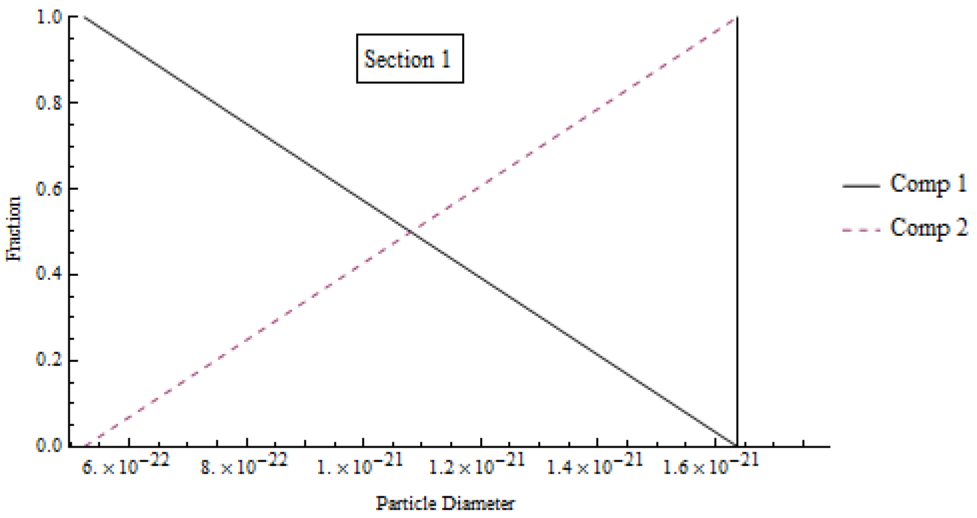


Figure 11 Initial component mass fraction distribution functions G_1 and G_2 of test scenario where $G_1(v) = (v - v_1)/(v_0 - v_1)$ and $G_2(v) = 1 - G_1(v)$. This serves as the initial aerosol component distribution of simulation DSMC_1.

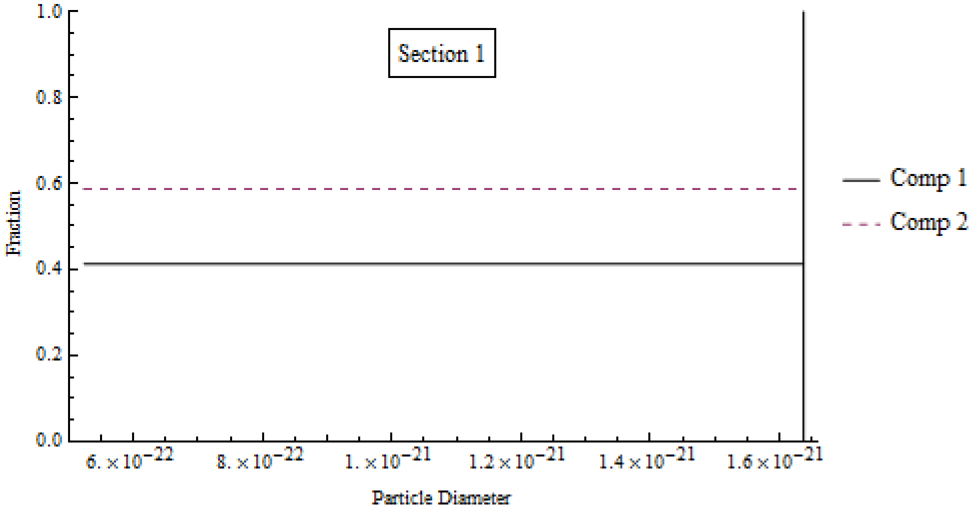


Figure 12 Component mass fraction as perceived by the sectional technique. Every particle within the section is assumed to have the same mass namely $m_{comp1} = N\int_{v_0}^{v_1} v\, G_1(v)dv$ and $m_{comp2} = N\int_{v_0}^{v_1} v\, G_2(v)dv$. This serves as the initial aerosol component distribution of simulation DSMC_2.

It is difficult to isolate the effect of this assumption so a comparison was made between two DSMC simulations. The first DSMC simulation, **DSMC_1**, is for a system of aerosols where the initial component distribution of each particle is assumed to be distributed as prescribed by figure 11 with a strong dependence of $G_k(v)$ on v. The second, **DSMC_2**, is for an aerosol where each particle has the same averaged component distribution as shown in 12. The initial conditions for both of these DSMC simulations assume an initial aerosol of 9.26×10^{12} particles/m^3 all having a diameter selected within the first size section in the range of $[10^{-8}, 1.46235 \times 10^{-8}]$ meters (the same sections as described in table II). The number distribution for the initial aerosol is selected to be constant versus particle mass over the section, hence corresponding to a value of $x = f(v) = v^2$ for the sectional technique. DSMC is computationally limited on the number of particles it can follow, so a sample list of 10^5 particles was created and a scale factor of 9.26×10^7 was used to rescale the results during and post computation. For each simulation, eight separate runs were completed and averaged for statistical purposes. Using the same initial conditions, a sectional simulation was also completed for comparison purposes, however, it is anticipated that this will deviate from both DSMC simulations rapidly due to the approximation on $x = f(v)$ as discussed earlier.

Of particular interest in this test scenario is how the function $\bar{v}_{\ell,k}$ differs between the two methods and the consequent effect it has as time progresses in the simulation. This function is computed from the sectional results by the equation $\bar{v}_{\ell,k}(v,t) = \frac{vQ_{\ell,k}(t)}{Q_\ell(t)}$. For the DSMC results, we calculate this function by, first, subdividing each section,ℓ, into 50 subsections,$(\ell_1, \ell_2, \dots, \ell_i, \dots, \ell_{50})$, and determining for every second of simulation time the amount of mass in each subsection per component. This will yield a discrete table of values of $Q_{\ell_i,k}$ averaged over eight runs. For each of these values, we calculate $\bar{v}_{\ell_i,k}(v,t)$ and perform a first order interpolation over v and time to create a continuous function. The simulation time was 100.0 seconds of particle interaction and plots of $\bar{v}_{\ell,k}$ at time $t = 0\ s$ in $\ell = 1$ are given in the following figures.

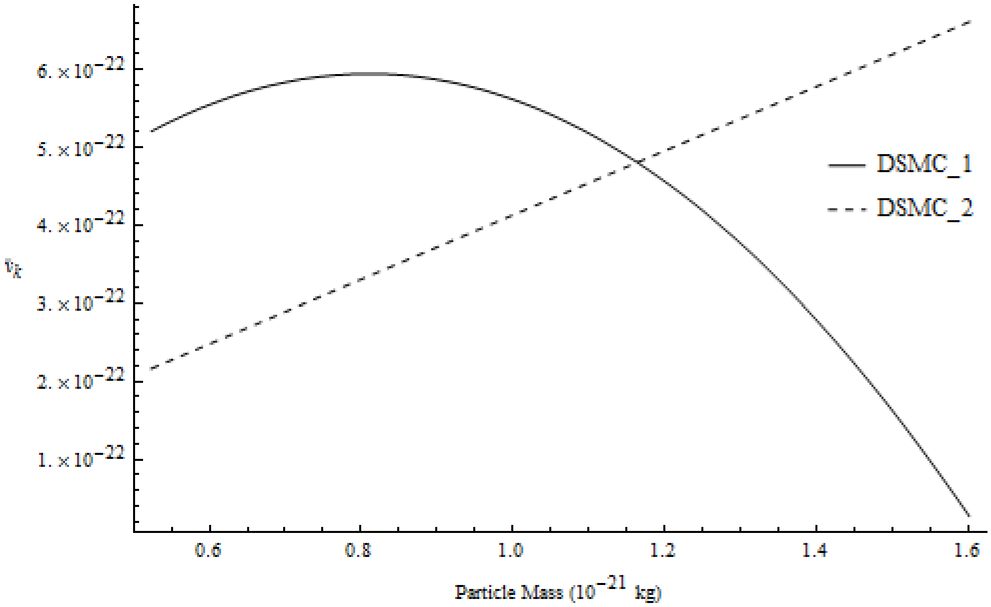


Figure 13 Plot comparing the value of $\bar{v}_{\ell,k}$ for $k = 1$ at time $t = 0$. sec for both DSMC simulations in the first size section.

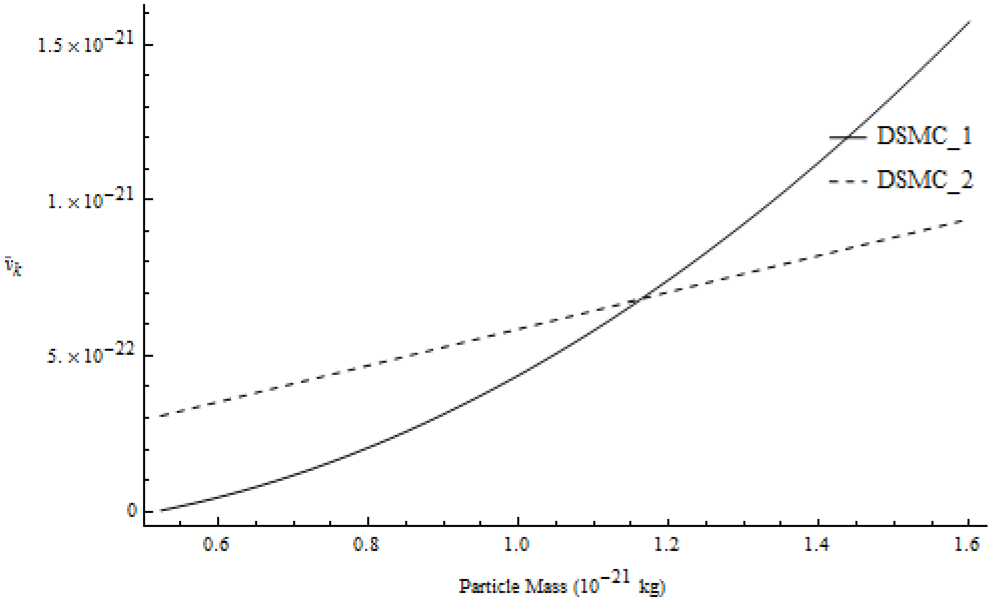


Figure 14 Plot comparing the value of $\bar{v}_{\ell,k}$ for $k = 2$ at time $t = 0$. sec for both DSMC simulations in the first size section.

Figures 13 and 14 reveal that there is a significant difference between the mean mass distribution $\bar{v}_{\ell,k}$ when considering an aerosol composed of averaged particles and those composed of a varying mass fraction even before time has evolved in the simulation. In tables VI- IX, we compare the results of the **DSMC_2** and sectional simulations against the results of **DSMC_1** at times $t = \{0.0, 1.0, 25.0, 100.0\}\ s$ for components one and two.

An important difference between the two simulations comes to light from analyzing the contents of these tables. In the case of **DSMC_1**, the smaller particles in section one have a much larger percentage of component one than the larger particles. It will take a great many of these smaller particles to coagulate into the next largest size section. Because of this, the mass of component one in section two will increase at a slower rate than is modeled by the averaged simulation, **DSMC_2** technique. This is the reason why **DSMC_2** shows a larger amount of component one in section two very early on in the simulation with the difference compounding in time. The exact opposite is true for component two since the larger particles have a greater amount of this component than the first simulation. This translates to **DSMC_2** showing a smaller amount of component two in section two than **DSMC_1** predicts. Here we calculate the percent difference by the formula $P.D. = (a - b)/((a + b)/2)$ where a is the base value chosen here to be the results of **DSMC_1**.

Table VI The initial mass per unit volume, $Q_{\ell,k}$, in the first two size sections for the sectional and DMC simulations for components one and two. All other sections are set to 0.

Section	Comp	DSMC_1	DSMC_2	Sec ($f(v)=v^2$)
1	1	$(4.134 \pm 0) \times 10^{-9}$	$(4.134 \pm 0) \times 10^{-9}$	4.134×10^{-9}
	2	$(5.866 \pm 0) \times 10^{-9}$	$(5.866 \pm 0) \times 10^{-9}$	5.866×10^{-9}
2	1	0.	0.	0
	2	0.	0.	0

Table VII A comparison of the mass represented in the first nine size sections for the sectional and DMC simulations at time $t = 1.0\ sec$ for both components one and two.

Section	Comp	DSMC_1	DSMC_2	% Diff	Sec ($f(v)$=v^2)	% Diff
1	1	$(4.0663 \pm 0.0003) \times 10^{-9}$	$(4.0592 \pm 0.0004) \times 10^{-9}$	-0.18	4.059×10^{-9}	-0.18
	2	$(5.7539 \pm 0.0007) \times 10^{-9}$	$(5.7598 \pm 0.0006) \times 10^{-9}$	0.1	5.760×10^{-9}	0.1
2	1	$(6.77 \pm 0.03) \times 10^{-11}$	$(7.48 \pm 0.04) \times 10^{-11}$	10.07	7.459×10^{-11}	9.73
	2	$(1.121 \pm 0.007) \times 10^{-10}$	$(1.062 \pm 0.006) \times 10^{-10}$	-5.4	1.058×10^{-10}	-5.73
3	1	0.	0.		3.855×10^{-13}	
	2	0.	0.		5.469×10^{-13}	
4	1	0.	0.		9.705×10^{-15}	
	2	0.	0.		1.377×10^{-14}	
5	1	0.	0.		1.901×10^{-16}	
	2	0.	0.		2.697×10^{-16}	
6	1	0.	0.		5.701×10^{-21}	
	2	0.	0.		8.090×10^{-21}	
7	1	0.	0.		6.711×10^{-26}	
	2	0.	0.		9.522×10^{-26}	
8	1	0.	0.		1.825×10^{-31}	
	2	0.	0.		2.590×10^{-31}	
9	1	0.	0.		0	
	2	0.	0.		0	

Table VIII A comparison of the mass represented in the first ten size sections for the sectional and DMC simulations at time $t = 25.0\ sec$ for both components one and two.

Section	Comp	DSMC_1	DSMC_2	% Diff	Sec ($f(v)=v^2$)	% Diff
1	1	$(2.8130 \pm 0.0013) \times 10^{-9}$	$(2.7140 \pm 0.0014) \times 10^{-9}$	−3.58	2.745×10^{-9}	−2.45
	2	$(3.756 \pm 0.002) \times 10^{-9}$	$(3.851 \pm 0.002) \times 10^{-9}$	2.5	3.895×10^{-9}	3.62
2	1	$(1.2910 \pm 0.0018) \times 10^{-9}$	$(1.379 \pm 0.002) \times 10^{-9}$	6.55	1.187×10^{-9}	−8.38
	2	$(2.046 \pm 0.003) \times 10^{-9}$	$(1.957 \pm 0.003) \times 10^{-9}$	−4.46	1.685×10^{-9}	−19.34
3	1	$(3.00 \pm 0.10) \times 10^{-11}$	$(4.12 \pm 0.09) \times 10^{-11}$	31.92	1.919×10^{-10}	146.12
	2	$(6.40 \pm 0.18) \times 10^{-11}$	$(5.80 \pm 0.13) \times 10^{-11}$	−9.27	2.723×10^{-10}	123.71
4	1	0.	0.		9.768×10^{-12}	
	2	0.	0.		1.386×10^{-11}	
5	1	0.	0.		2.286×10^{-13}	
	2	0.	0.		3.244×10^{-13}	
6	1	0.	0.		1.661×10^{-15}	
	2	0.	0.		2.357×10^{-15}	
7	1	0.	0.		5.722×10^{-18}	
	2	0.	0.		8.119×10^{-18}	
8	1	0.	0.		4.571×10^{-21}	
	2	0.	0.		6.487×10^{-21}	
9	1	0.	0.		2.395×10^{-24}	
	2	0.	0.		3.398×10^{-24}	
10	1	0.	0.		5.064×10^{-30}	
	2	0.	0.		7.185×10^{-30}	

Table IX A comparison of the mass represented in the first ten size sections for the sectional and DMC simulations at time $t = 100.\ sec$ for both components one and two.

Section	Comp	DSMC_1	DSMC_2	% Diff	Sec ($f(v)=v^2$)	% Diff
1	1	$(1.1550 \pm 0.0012) \times 10^{-9}$	$(1.0700 \pm 0.0014) \times 10^{-9}$	−7.66	1.164×10^{-9}	0.78
	2	$(1.434 \pm 0.003) \times 10^{-9}$	$(1.518 \pm 0.002) \times 10^{-9}$	5.67	1.652×10^{-9}	14.09
2	1	$(2.156 \pm 0.004) \times 10^{-9}$	$(2.169 \pm 0.004) \times 10^{-9}$	0.62	1.670×10^{-9}	−25.4
	2	$(3.083 \pm 0.004) \times 10^{-9}$	$(3.078 \pm 0.005) \times 10^{-9}$	−0.17	2.370×10^{-9}	−26.17
3	1	$(8.18 \pm 0.03) \times 10^{-10}$	$(8.89 \pm 0.04) \times 10^{-10}$	8.39	1.060×10^{-9}	25.82
	2	$(1.339 \pm 0.004) \times 10^{-9}$	$(1.262 \pm 0.006) \times 10^{-9}$	−5.95	1.504×10^{-9}	11.61
4	1	$(5.6 \pm 0.8) \times 10^{-12}$	$(5.8 \pm 0.5) \times 10^{-12}$	3.71	2.239×10^{-10}	190.26
	2	$(9.0 \pm 1.2) \times 10^{-12}$	$(8.2 \pm 0.7) \times 10^{-12}$	−13.71	3.177×10^{-10}	188.46
5	1	0.	0.		1.570×10^{-11}	
	2	0.	0.		2.227×10^{-11}	
6	1	0.	0.		4.077×10^{-13}	
	2	0.	0.		5.786×10^{-13}	
7	1	0.	0.		4.340×10^{-15}	
	2	0.	0.		6.159×10^{-15}	
8	1	0.	0.		2.002×10^{-17}	
	2	0.	0.		2.841×10^{-17}	
9	1	0.	0.		4.155×10^{-20}	
	2	0.	0.		5.895×10^{-20}	
10	1	0.	0.		4.050×10^{-23}	
	2	0.	0.		5.747×10^{-23}	

As time progresses in the simulation, note that the percent difference between simulations **DSMC_1** and **DSMC_2** decreases for both components one and two. Over time, enough coagulation has occurred for there to be sufficient mixing of the particles and the component distribution has averaged out to be similar to that of the sectional approximation. This would seem to suggest that as time progresses, the divergence of the sectional technique from actual particle physics becomes less pronounced. However, recall that the disagreement brought on by the approximation on the number distribution discussed earlier becomes *more* pronounced in time. Hence we find that the second approximation of this technique calls into question results early on in the simulation while the first approximation does the same as time evolves. Of course, by selecting a greater number of sections and subdividing the mass intervals further, the consequences of both of these approximations will be lessened. However, depending on the shape of the initial number distribution and the component distribution, a large number of subdivisions may be needed to accurately model the aerosol which would greatly increase computational cost.

In our discussion in this section, we operated under the assumption that a component distribution is known but is not being utilized by the sectional technique. We should note, however, that often detailed information on the aerosol component distribution may not be available. In that case, it may be advantageous to use approximations such as those of the sectional technique which require only the mean mass fraction and not a specified component distribution. Similarly, use of DSMC may be disadvantageous if the component distribution is not known and an assumption must then be made. There is only a loss of information by the sectional technique if that information was available to begin with.

VII. Explorations in Coupling

VII.A. Method One

Since DSMC provides fidelity to multicomponent effects of an aerosol, it was thought that perhaps by deriving the more precise values of $\bar{v}_{\ell,k}$ from DSMC, they could be inserted into the sectional technique to improve its accuracy as well. In this way, one could take advantage of the computational speed of a sectional technique such as MAEROS while still maintaining fidelity to the particle multiphysics that DSMC provides. The nature of this coupling would be to perform a simplified DSMC simulation in parallel with the sectional technique and use the calculated values of $\bar{v}_{\ell,k}$ to remove the second approximation described in the previous section on the mass distribution.

The sectional technique assumes that the "mean mass fraction of all particles within a section [is] equal and thus independent of particle size within a section.[16]" To improve upon this, then, we need the mean mass fraction to vary within a given section. This would provide more refined results and eliminate the strong approximations over the components. In other words, we would like for $\bar{v}_{\ell,k}$to be a continuous function of v within a given section. This can be accomplished by performing a DSMC simulation, calculating the values of $\bar{v}_{\ell,k}$ over time from the resultant data, and then, as was achieved in the test scenario in the previous section, create an interpolation over time and particle size of the discrete values DSMC provides. This function of $\bar{v}_{\ell,k}$ is then a function of both time and v and is incorporated into the sectional equations as follows.

Keeping the first approximation of $n(v,t) = \frac{Q_\ell f'(v)}{v(x_\ell - x_{\ell-1})}$ and adding this new value for $\bar{v}_{\ell,k}$, we construct a differential equation for $Q_{\ell,k}(t)$. In keeping with the notation of Gelbard,[16] this equation becomes

$$\frac{dQ_{\ell,k}}{dt} = \frac{1}{2}\sum_{i=1}^{\ell-1}\sum_{j=1}^{\ell-1}[\bar{\beta}_{ij\ell}^{(1a)}Q_jQ_i + \bar{\beta}_{ij\ell}^{(1b)}Q_jQ_i] - \sum_{i=1}^{\ell-1}[\bar{\beta}_{i\ell}^{(2a)}Q_\ell Q_i - \bar{\beta}_{i\ell}^{(2b)}Q_\ell Q_i] \\ - \frac{1}{2}\bar{\beta}_{\ell}^{(3)}Q_\ell Q_\ell - Q_\ell \sum_{i=\ell+1}^{m}\bar{\beta}_{i,\ell}^{(4)}Q_i \tag{25}$$

where the beta values have necessarily been redefined as

$$\begin{aligned}
\bar{\beta}_{i,j,\ell,k}^{(1a)} &= \int_{x_{i-1}}^{x_i}\int_{y_{j-1}}^{y_j} \frac{\theta(v_{\ell-1} < u+v < v_\ell)\,\bar{u}_{\ell,k}(v,t)\,\beta(u,v)}{u\,v\,(x_i - x_{i-1})(y_i - y_{i-1})}dydx \\
\bar{\beta}_{i,j,\ell,k}^{(1b)} &= \int_{x_{i-1}}^{x_i}\int_{y_{j-1}}^{y_j} \frac{\theta(v_{\ell-1} < u+v < v_\ell)\;\bar{v}_{\ell,k}(v,t)\,\beta(u,v)}{u\,v\,(x_i - x_{i-1})(y_i - y_{i-1})}dydx \\
\bar{\beta}_{i,\ell,k}^{(2a)} &= \int_{x_{i-1}}^{x_i}\int_{y_{j-1}}^{y_j} \frac{\theta(u+v > v_\ell)\;\bar{u}_{\ell,k}(v,t)\,\beta(u,v)}{u\,v\,(x_i - x_{i-1})(y_\ell - y_{\ell-1})}dydx \\
\bar{\beta}_{i,\ell,k}^{(2b)} &= \int_{x_{i-1}}^{x_i}\int_{y_{j-1}}^{y_j} \frac{\theta(u+v < v_\ell)\,\bar{v}_{\ell,k}(v,t)\beta(u,v)}{u\,v\,(x_i - x_{i-1})(y_\ell - y_{\ell-1})}dydx \\
\bar{\beta}_{\ell,k}^{(3)} &= \int_{x_{\ell-1}}^{x_\ell}\int_{y_{\ell-1}}^{y_\ell} \frac{\theta(u+v > v_\ell)\big[\,\bar{u}_{\ell,k}(v,t)\; + \bar{v}_{\ell,k}(v,t)\big]\beta(u,v)}{u\,v\,(x_\ell - x_{\ell-1})(y_\ell - y_{\ell-1})}dydx \\
\bar{\beta}_{i,\ell,k}^{(4)} &= \int_{x_{i-1}}^{x_i}\int_{y_{j-1}}^{y_j} \frac{\bar{u}_{\ell,k}(v,t)\,\beta(u,v)}{u\,v\,(x_\ell - x_{\ell-1})(y_\ell - y_{\ell-1})}dydx.
\end{aligned} \tag{26}$$

Note that for each of these beta values there is now the presence of the functions $\bar{v}_{\ell,k}(v,t)$ and $\bar{u}_{\ell,k}(v,t)$. These cannot be pulled out of the integrals, as in the case of the original version of the sectional technique, because of their dependence on v. As a result, the beta values are now functions of time as well as species. Recall that what made the original sectional technique computationally feasible was that the beta values were independent of species and time and could hence be computed once for each section and then kept as constants within a differential solver. In this current form, not only must each beta be calculated k_{total} times, but they can no longer be constant coefficients within the differential equation and must be evaluated at every time step in the numeric differential equation solver as well. This would increase the computation time drastically to the point that it is entirely impractical. The user could perform several thorough DSMC simulations by the time the coupled simulation was complete. For this reason, coupling these programs in this manner is not computationally feasible.

VII.B. Method Two

Another means of coupling considered was to create a hybrid program which would dynamically switch between the DSMC and sectional methods depending upon the importance of the multicomponent aspect of the aerosol. Two conversion programs were constructed to change data formats from sectional to DSMC and vice versa.

The conversion from a list of particles to a mass distribution is a straight-forward process. The DSMC result list is sorted according to the total mass of each particle from least to greatest. A bin count is then performed to determine the number of particles which have a diameter within each size section, ℓ. Using these values, the amount of sample mass per section and component is calculated and multiplied by the mass scale factor, f, in order to give the actual mass per section. These values can then be entered into the sectional program as initial conditions of the form $q[\ell][k][t=0]$.

To convert from a sectional distribution is a more involved process. This program first calculates the distribution of mass over each size section in a table of the form

$$\{p_1, p_2, \dots, p_{\ell\text{total}}\}.$$

Also, the component distribution for each given section, ℓ, is arranged in a table of the form

$$\{p_{\ell_1}, p_{\ell_2}, \dots, p_{\ell_{\mathrm{ktotal}}}\}\ .$$

Once this is determined, the program then takes the number of sample particles that the user would like to feed into the DSMC program and determines the overall mass this represents in each section if the particles were of mean size. The total sample mass is then redistributed over the sections according to the original sectional proportion and put into a sample mass distribution table. Once the sample mass for each section is determined, this information is then fed to a particle source module. This module generates a list of particles within the section of random diameter where each particle has the same component distribution that was calculated for that section. The module generates as many particles to match as close as possible the sample mass it was fed without going over. Figure 15 shows a diagram of how a particle is selected for a given size section. Note that all particles within a size section have the same component distribution.

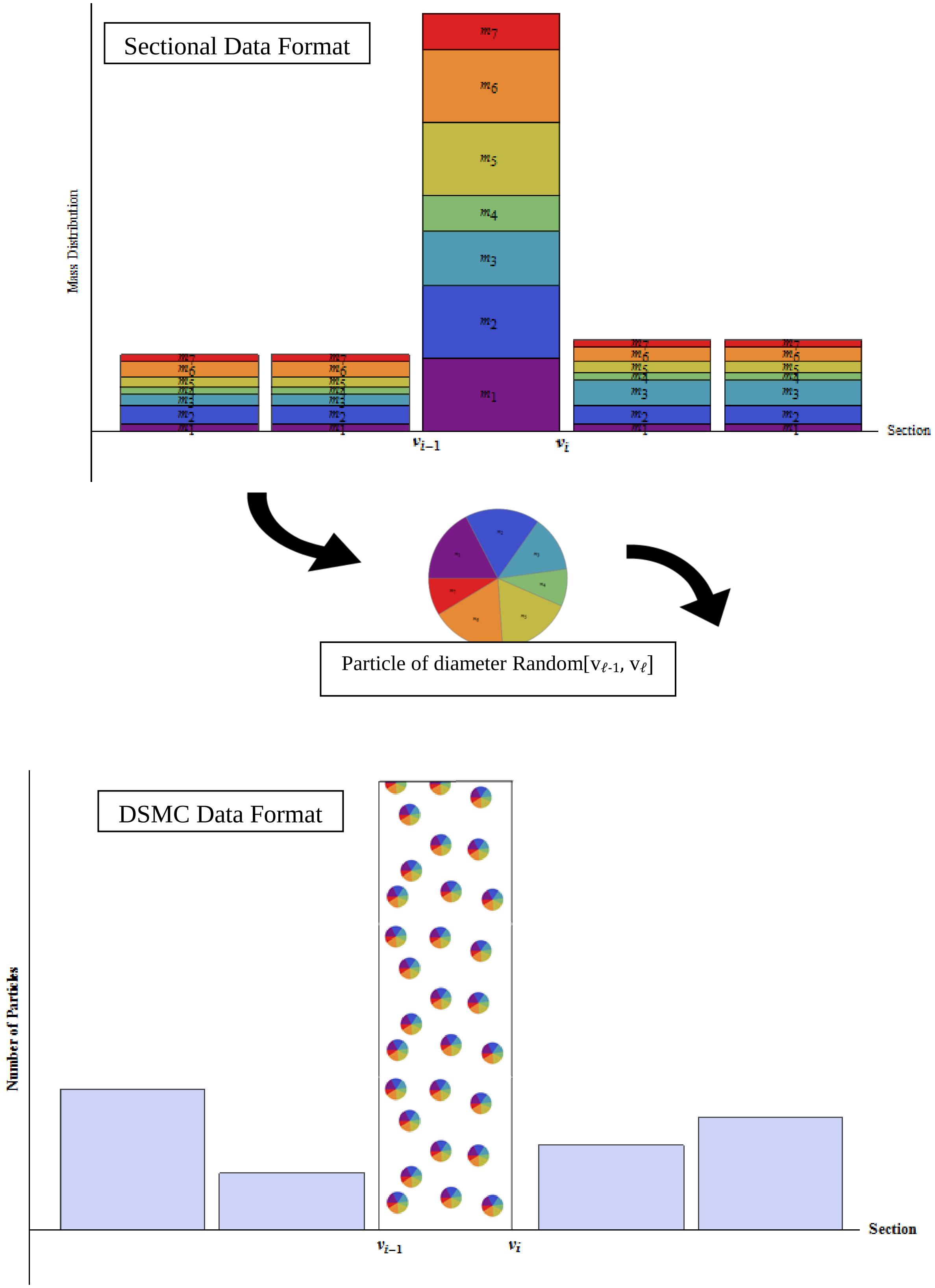
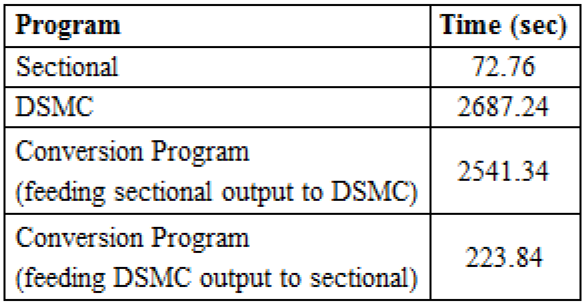


Figure 15 Diagram of sectional to DSMC conversion process

At first glance, this method of coupling seems to be a straightforward means of making use of the speed of the sectional technique when possible and the DSMC accuracy when needed. However, the computational expense of the aforementioned conversion process makes this a less attractive option. To demonstrate, the initial conditions from the MAEROS user manual test problem[17] (refer to table II) were fed into the sectional program for a simulation time of 100s with solely coagulation affecting the aerosol. These sectional initial conditions were converted to a DSMC format with initially 10^6 simulated particles. This data was also run by DSMC and its final result at $t = 100$s was converted back to a sectional format. The run times of each program are given in the table below. The time for conversion from DSMC to sectional was for a single conversion. In practice, one would have as many as ten DSMC simulations being performed for statistical purposes and each would need to be converted. There is then little to no savings in computation time to switch from DSMC to sectional except perhaps for a much lengthier simulation. Also note that the time taken to convert from a sectional to DSMC format is nearly as long as performing the DSMC simulation itself. Clearly this method of coupling is also computationally impractical.

Table X Timing of the test problem.



Program	Time (sec)
Sectional	72.76
DSMC	2687.24
Conversion Program (feeding sectional output to DSMC)	2541.34
Conversion Program (feeding DSMC output to sectional)	223.84

VIII. DSMC Improvements on Computational Speed

A large portion of the computation time in DSMC is devoted to calculating the coagulation coefficient $\beta(u,v)$ for a pair of particles about to undergo coagulation. While it is a fairly simple calculation, numerous variables are introduced and manipulated in the process. Within Mathematica, this can be time consuming as it must determine the nature of these variables (number, list, algebraic object, etc.) each time it is called. To bypass this expensive process, we have compiled this portion of the code and hence pre-defined the parameters of all input variables. Computation time decreased substantially with the implementation of this compiled code alone.

Another process that can be computationally intensive in DSMC is the list manipulation. In the DSMC codes of previous works[5-9], within each time step, Δt_i, a certain number of particle pairs, p_i, is tested for coagulation by sampling the whole particle list, checking if coagulation has occurred, and then, if confirmed, redefining the list with the two particles summed together. With this method, there could be as many as p_i redefinitions of the particle list within the time step Δt_i,. Alternatively, we have now written the code so that within the time $\mathit{\Delta} t_i$, we check all p_i pairs at once for possible collisions and then adjust the list once at the end of the time step. This process allows for a single redefinition of the list in each time step.

By implementing both of these processes together, we have been able to decrease the overall computation time of DSMC by over thirty times. A simulation that originally took an hour to compute is being performed in a matter of minutes.

IX. Conclusion

By considering a simple example for a single component problem for which analytical solutions are available, we have found that the sectional method results are quite sensitive to the assumptions on the intra-sectional number distribution and the choice of the function, $f(v)$. Therefore, the choice in this function should not be arbitrary, but carefully selected based on the distribution of initial data.

For a multicomponent aerosol, we have found that with the further assumption on the intra-sectional component distribution, the sectional results can differ substantially from the DSMC results. The two DSMC simulations further demonstrated the effect of sectional assumptions early in the simulation.

We explored coupling of the sectional and DSMC methods in the effort to improve upon the accuracy of the sectional technique. Our efforts to date have not been successful as more computational time was spent in carrying out the coupling as compared to computational time needed with DSMC alone. Obviously, we have not exhausted all possibilities, and there still may be ways to carry out an effective coupling technique.

Within the framework of the computational program Mathematica, we have been successful in increasing the speed of the DSMC program through various means. While the computational speed of this faster program is still far slower than that of the corresponding sectional programs, ways to make the DSMC computations still faster should be further explored so that the method can become practical for use in large scale nuclear source term programs. Additionally, it will, of course, be of considerable interest to verify DSMC computations against experimental data. Such data could be obtained, for example, with the apparatus and techniques recently reported.[22]

Finally, we have confined our attention here to coupling the sectional method with DSMC because of the use of the sectional method in CONTAIN and MELCOR, but other methods such as the moments or finite elements are also in use in codes such as NAUA and VICTORIA.[4,23] It would be worthwhile to explore coupling of DSMC with these methods as well.

Acknowledgements

This research was supported by a Department of Energy grant (NEUP-964). S. Campbell was also supported by a U.S. Department of Education GAANN fellowship and a U.S. Nuclear Regulatory Commission graduate fellowship. We would like to thank the reviewers of this paper for their helpful suggestions.

References:

1. K. K. MURATA D. C. WILLIAMS, J. TILLS, R. O. GRIFFITH, R. G. GIDO, E. L. TADIOS, F. J. DAVIS, G. M. MARTINEZ, K. E. WASHINGTON, "Code Manual for CONTAIN 2.0: A Computer Code for Nuclear Reactor Containment Analysis," Sandia National Laboratories, Albuquerque, NM, Prepared for the U.S. Nuclear Regulatory Commission, NUREG/CR-6533, SAND97-1735 (1997)
2. R. O. GAUNTT, R. K. COLE, C. M. ERICKSON, R. G. GIDO, R. D. GASSER, S. B. RODRIGUEZ, M. F. YOUNG, "MELCOR Computer Code Manuals," Sandia National Laboratories, Albuquerque, NM, Prepared for the U.S. Nuclear Regulatory Commission, NUREG/CR-6119, SAND2000-2417/1 (2000)
3. M. M. R. WILLIAMS and S. K. LOYALKA, "Aerosol Science: Theory and Practice with Special Applications to the Nuclear Industry," Pergamon, Oxford (1991).
4. R. SHER and R. R. HOBBINS, "Transport and Removal of Aerosols in Nuclear Power Plants Following Severe Accidents," American Nuclear Society (2011)
5. S. K. LOYALKA "Direct Simulation of Multi-Component Aerosol Dynamics," *Trans. Am. Nucl. Soc.,* **88**, 334 (2003)
6. G. PALANISWAAMY and S. K. LOYALKA, "Direct Simulation Monte Carlo Multi-Component Aerosol Dynamics: Coagulation, Deposition and Source Reinforcement," *Nucl. Tech.,* **160**(2), 187-204 (2007)
7. G. PALANISWAAMY and S. K. LOYALKA, "Direct Simulation Monte Carlo Aerosol Dynamics: Coagulation and Condensation," *Ann. Nucl. Energy,* **35**(3), 485-494 (2008)

8. G. PALANISWAAMY and S. K. LOYALKA, "Direct Simulation Monte Carlo Aerosol Dynamics: Collisional Sampling Algorithms," *Ann. Nucl. Energy* **34**(1), 13-21 (2007)
9. J. PALSMEIER and S.K. LOYALKA, "Evolution of Charged Aerosols: Role of Charge on Coagulation," *Nucl. Tech.*, **184**(1), 78-95 (2013)
10. Y. EFENDIEV and M. R. ZACHARIAH, "Hybrid Monte Carlo Method for Simulation of Two-Component Aerosol Coagulation and Phase Segregation," *J. of Colloid. and Interface Sci.*, **249**, 30-43 (2002)
11. Z. SUN, R. L. AXELBAUM, and J. I. HUERTAS, "Monte Carlo Simulation of Multicomponent Aerosols Undergoing Simultaneous Coagulation and Condensation," *Aerosol Sci. and Tech.*, **38**(10), 963–971 (2004)
12. M. SMITH, K. LEE and T. MATSOUKAS, "Coagulation of charged aerosols," *J. of Nanoparticle Res.*, **1**, 185–195 (1999).
13. J. WEI, "A Monte Carlo Method for Coagulation of Charged Particles," *J. of Aerosol Sci.*, **65**, 21-25 (2013)
14. N. A. FUCHS, "The Mechanics of Aerosols," Revised ed., Pergamon Press New York (1964)
15. S. K. FRIEDLANDER, "Smoke, Dust, and Haze: Fundamentals of Aerosol Dynamics," 2nd ed., New York: Oxford University Press, (2000).
16. F. GELBARD, and J. H. SEINFELD, "Simulation of Multicomponent Aerosol Dynamics," *J. of Colloid. and Interface Sci.*, **78**(2), 485-501 (1980)
17. F. GELBARD, "MAEROS User Manual", Sandia National Laboratories, NUREG/CR-1391, SAND 80-0822 (1982)

18. F. GELBARD, “Modeling Multicomponent Aerosol Particle Growth by Vapor Condensation,” *Aerosol Sci. and Tech.*, **12**(2), 399-412 (1990)

19. G. A. BIRD, “Molecular Gas Dynamics and Direct Simulation of Gas Flows,” 2nd ed. Oxford University Press, London (1994).

20. W. SCOTT, “Analytic Studies of Cloud Droplet Coalescence I,” *J. of the Atmospheric Sci.*, **25**, 54-65 (1968)

21. A. A. LUSHNIKOV, "Evolution of coagulating systems. III. Coagulating mixtures," *J. Colloid and Interface Sci.*, **54**, 94 (1976)

22. M. P. Simones and S. K. Loyalka, “Measurements of Charged Aerosol Coagulation,” *Nuc. Tech.*, **189**, 45-62 (2015)

23. N. E. BIXLER, “VICTORIA 2.0: A Mechanistic Model for Radionuclide Behavior in a Nuclear Reactor Coolant System Under Severe Accident Conditions,” Sandia National Laboratories, Albuquerque, New Mexico, Prepared for the U.S. Nuclear Regulatory Commission, NUREG/CR-6131, SAND93-2301 (1998)

DRAFT of a Manuscript

Mesh Free Simulation of Aerosol Evolution

Shawn Campbell, John Palsmeier, and Sudarshan K. Loyalka*

Nuclear Science and Engineering Institute &
Particulate Systems Research Center
E2433 Lafferre Hall
University of Missouri
Columbia, MO 65211

*Corresponding Author

Sudarshan K. Loyalka
loyalkaS@missouri.edu
+ 1 (573) 882-3568
E2433 Lafferre Hall
University of Missouri
Columbia, MO 65211

Total Number of Pages: 32
Total Number of Tables: 6
Total Number of Figures: 8

Abstract

The nuclear source term is greatly affected by the formation and presence of aerosols in the reactor primary vessel and the containment. In simulations, the aerosol distribution is often assumed spatially homogeneous (well-mixed), and there have been relatively few studies of the effects of spatial inhomogeneity on aerosol evolution in nuclear accidents.   We have explored here an extension of some our recent work on the Direct Simulation Monte Carlo Method (DSMC) to spatially inhomogeneous aerosol. In doing so, we have also departed from the traditional applications of the DSMC method where the computational domain is divided into fixed cells. We have explored here an alternative, mesh-free method by utilizing a clustering technique. This technique associates particles according to a distance parameter and is commonly used in group theory and machine learning. To benchmark this mesh-free modeling, we have verified the DSMC results against those obtained from the use of cell balanced sectional technique for a spherical geometry where both coagulation and diffusion take place.

I. Introduction

The nuclear source term is greatly affected by the formation and presence of aerosols in the reactor primary vessel and the containment [1-8]. While considerable progress has been made in both modeling and experimental measurements of aerosol evolution [ ], there remain many challenges of both fundamental and applied nature. For example, the aerosol distribution is generally assumed spatially homogeneous (well-mixed), and there have been relatively few studies [ ] of the effects of spatial inhomogeniety on aerosol evolution in nuclear accidents.

It is our purpose in this paper to explore such an extension of some of our recent work with the Direct Simulation Monte Carl Method (DSMC). In doing so, we have also departed from the traditional applications of the DSMC method where the computational domain is divided into fixed cells. There, in each time step, the particle positions are evolved in time and those within a given cell (or mesh) are assessed for interaction using the No Time Counter (NTC) method [ ]. In his book, Bird [ ] alludes to an alternative to regional meshes in cases where the construction of a mesh can be a labor intensive and computationally expensive process. This approach creates a distribution of points within the geometry and then uses a reference scheme in which a particle “is said to be ‘in a cell’ when it is nearest a point which specifies that cell. The point reference scheme avoids the necessity of providing an analytical description of the cell boundaries and is, therefore, particularly useful for multi-dimensional problems with geometrically complex boundaries.”

We have explored an alternative, mesh-free method for modeling a spatially inhomogeneous (or heterogeneous) aerosol with DSMC by utilizing a clustering technique. This technique associates particles according to a distance parameter and is commonly used in group theory and machine learning. Much like the point reference scheme described by Bird, the

Partioning Around the Medoid (PAM) method [ ] associates each particle to its nearest representative point (medoids) in order to determine which particles will be sampled for interaction. However, in this case the medoids are determined in each time step by an optimizing algorithm that selects the points which minimize the "cost" in particle distance. As we seek to demonstrate, this adaptive technique can be used in place of a mesh to model aerosol interaction.

In section II, we give a detailed description of the DSMC technique and how the clustering algorithm is incorporated; with a description of how the clustering algorithm works provided in section III and how the diffusion step is calculated in section IV. To benchmark this mesh-free modeling, we describe in section V volumetric cell balanced sectional equations for a spherical geometry. Section VI describes the benchmark simulation and provides an analysis of the results. Finally we discuss our conclusions in section VII.

We note here that in the construction of this DSMC clustering program and the benchmark simulations we have used Wolfram Mathematica in the development and implementation. The choice was based on its built in clustering algorithms and features such as symbolic representation, listability, and data visualization.

II. DSMC Clustering Technique

Traditionally, in the case of a heterogeneous system, a mesh is applied over the space and, for every simulated time step, each particle's location is advanced and then grouped according to the mesh cell it happens to be in [ ]. The method explored here instead groups the particles according to a given parameter such as relative distance (for our purposes, Euclidean distance) using a clustering algorithm where each time the particles are allowed to move, the list is re-clustered allowing the groupings to adapt to the particle dynamics in time. Unlike a typical mesh, these clusters depend on the system geometry only indirectly and thus are particularly useful in the case of arbitrary geometries. An explanation of the clustering algorithm is given in the next section.

An outline of the steps involved in the proposed DSMC cluster method is given here with the primary difference between it and the original DSMC found in the clustering step 2c. For a given simulation:

1. Given the initial particle mass distribution, N particles are sampled and placed in an initial particle list with each entry p of the form $\{m_{p,1}, m_{p,2}, \dots, m_{p,s}, x_p, y_p, z_p\}$ where $m_{p,1}, m_{p,2}, \dots, m_{p,s}$ represent the masses of components 1 through s and x_p, y_p, z_p are the Cartesian coordinates of particle p sampled from the aerosol coordinate distribution.
2. For each time step Δt, the following processes take place:
    a. Each particle's position is evolved. Possible causes of movement are Brownian motion, gravitational settling, and bulk fluid flow. Details of this step are given in a later section.
    b. If a particle comes in contact with a surface, the particle may deposit or remain suspended. For the simulations in this work, it is assumed that all particles which come in contact with a surface deposit in that location and are thus removed from the simulation.
    c. The clustering algorithm PAM is used to partition the particle list into n_c clusters. This algorithm is described in the next section.
    d. Within each cluster, No Time Counter (NTC) method is used to determine the number of particle to particle interactions (coagulations) which have occurred.

Particles found to interact are combined as a single particle maintaining conservation of mass. The location of the new particle is taken to be a weighted, random choice between the original two particle positions.

   e. The deposited particles and corresponding locations are recorded and, if desired, the overall particle list can be exported for later analysis.

3. The current time is incremented by Δt and step two is repeated until the desired end time is reached.

When two particles, p_1 and p_2, are determined to have coagulated (step 2d), a new particle is formed and its position is taken to be a weighted, random choice between the original particles' locations. The mean displacement via Brownian diffusion of a particle is given by $\Delta x_p = \sqrt{2D\Delta t}$ for

$$D_p = \frac{k_B T}{3\pi d_p \mu} \cdot C_c \tag{1}$$

where k_B is Boltzmann's constant, T is the temperature, d_p is the particle diameter, μ is the air viscosity, and C_c is the Cunningham correction factor. Hence, the displacement of a particle is approximately proportional to $m^{1/6}$. In a time step, then, coordinates one and two are weighted according to the mass of the individual particles to be $W_1 = 1/(1+w)$ and $W_2 = w/(1+w)$ where $w = \left(\frac{m_2}{m_1}\right)^{1/6}$ and a choice is made is made between them using these weights. The alternative to this would be to select an averaged location based on the center of mass but this can have a long term effect on the particle distribution. As time passes in the simulation, the net effect is that the particles move towards the center of the geometry. For the purpose of this work, the random choice option was selected to preserve the existing particle distribution at each time step.

The time step Δt is chosen so that the number of pairs selected for possible coagulation (Bird's NTC method)# within a cluster,

$$pairs = \frac{1}{2} N^2 \beta \Delta t / V \tag{2}$$

is small compared with the average collision rate of the particles[15] (#), where β is the maximal coagulation kernel, and V is the cluster volume. Within this work, the time step has been chosen such that the number of particle pairs interacting within that time is less than one percent of the total number of particles. This ensures the time step is reasonably small and provides an upper bound on our simulation time step which is dependent on the number density of the aerosol and the sample list length N.

In figure 1, we illustrate with a random distribution of points what a typical cluster versus triangular mesh grouping would be in two dimensions. This comparison brings to light an important aspect of the clustering technique that must be rectified during implementation, namely, the empty space around each cluster. The volume of the clusters is computed by forming a convex hull around the points. Since the volume ends at the outermost points then, it can be significantly less than that of the cells in a mesh. When performing the NTC method of DSMC, this will affect the number of particle pairs selected for interaction since the ratio of particle cross-sectional volume to the cell volume will be greater. For this work, this is accounted for each cluster by volumetrically scaling each one so that the sum of the volumes encompasses the entire geometric volume. In this way, there is no volume lost among the clusters. This is an area for which further investigation could be beneficial to determine the optimal way to handle the space between clusters.

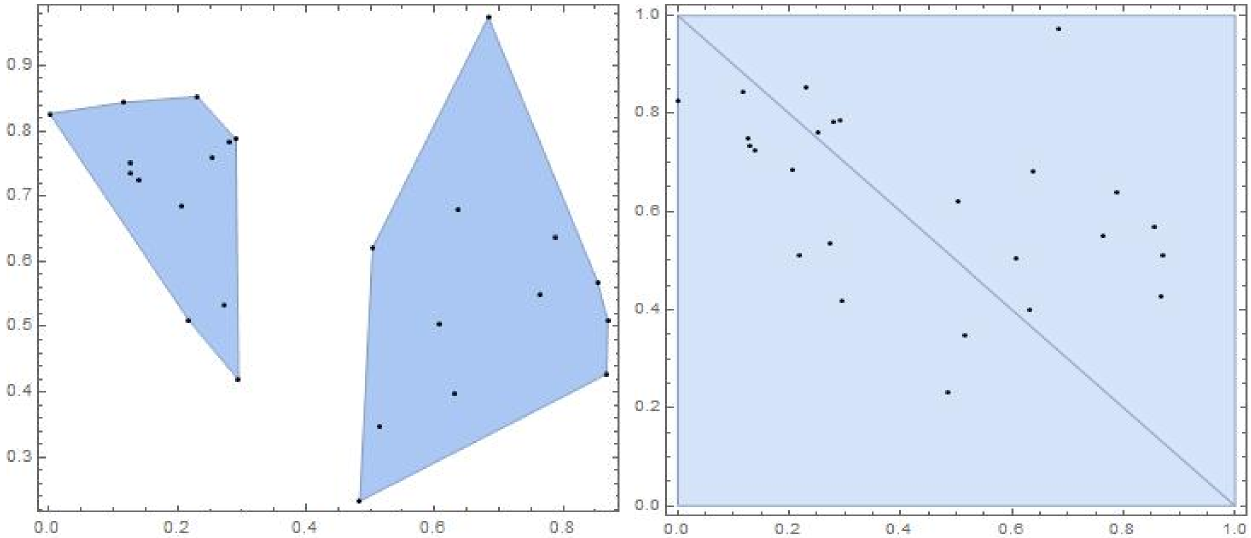


Figure 1 Given a distribution of points within a simple square geometry, the figure on the left demonstrates two groups formed by the clustering algorithm while the figure on the right demonstrates groups formed by an arbitrary mesh.

When applying DSMC over a mesh, the user has direct control over the position of the cell boundaries. Therefore, the cell locations and sizes are typically chosen in such a way that the change in flow properties of the aerosols is minimized among the cells [   ]. In contrast, the cluster technique creates associations based on the particle's instantaneous relative positions at each time step with the primary control the user maintains over the clusters being the number of clusters used. By carefully selecting the number of clusters n_c, one can control the size and average number of elements each contains.

Determining the number of clusters for a given simulation is sensitive to a number of parameters and various algorithms exist for determining its optimal value. However, a general rule of thumb for the number of clusters is taken to be [ ].

$$n_c = \sqrt{N/2}. \tag{3}$$

However, for purposes of this method, the optimal number of clusters is actually dependent on the time step, the size of the containing geometry, and the mobility of the particles. For the cluster technique to be valid within DSMC, the majority of particles within a given cluster should not have left that cluster during the time step Δt. Otherwise, the assumption that only particles within a cluster interacted with one another becomes invalid. With this in mind, the number of clusters, n_c, is chosen so that the clusters are large enough that the mean displacement Δx of the contained particles is $1/100$ the cluster diameter d_c. If we assume spherical clusters, this very roughly translates to $n_c = \frac{6\,V}{\pi(100\,\Delta x)^3}$ where V is the containing geometry volume and Δx is the mean displacement of an average particle in time Δt. For the simulations in this work in which particles are assumed to diffuse solely due to Brownian motion, $\Delta x = \sqrt{2\,D\,\Delta t}$ with D being the average diffusion coefficient of the particle list. Hence we have

$$n_c = \frac{6\,V}{\pi\left(100\,\sqrt{2\,D\,\Delta t}\right)^3} \tag{4}$$

as another approximate rule of thumb. Within the clustering program, the number of clusters is taken to be the minimum of equations 3** and 4**. It is important to note that equation 4** is highly dependent on the particle size and mobility. To illustrate this, consider an aerosol distribution within a sphere of radius $R = .05$m like that given in section VI. Figure 2 demonstrates the value of n_c for particle radii in the range of $0.001 - 0.01\ \mu m$ (in the free molecular regime). The number of clusters for such an aerosol is greatly limited since the particles are highly mobile and it is difficult to restrict them to a cluster in a time step.

Conversely, if the size of the clusters is too large, then a problem could arise from particles at opposite ends being chosen for interaction and a propagation of systematic error with respect to the particle distribution. Bird accounts for this in the case of a mesh by creating a sub-mesh over each cell [ ] . In the same way, a sub-cluster could be generated over these overly large clusters to ensure only neighboring particles interact. Further work is needed to ascertain a method for determining the optimal value of n_c for a given simulation.

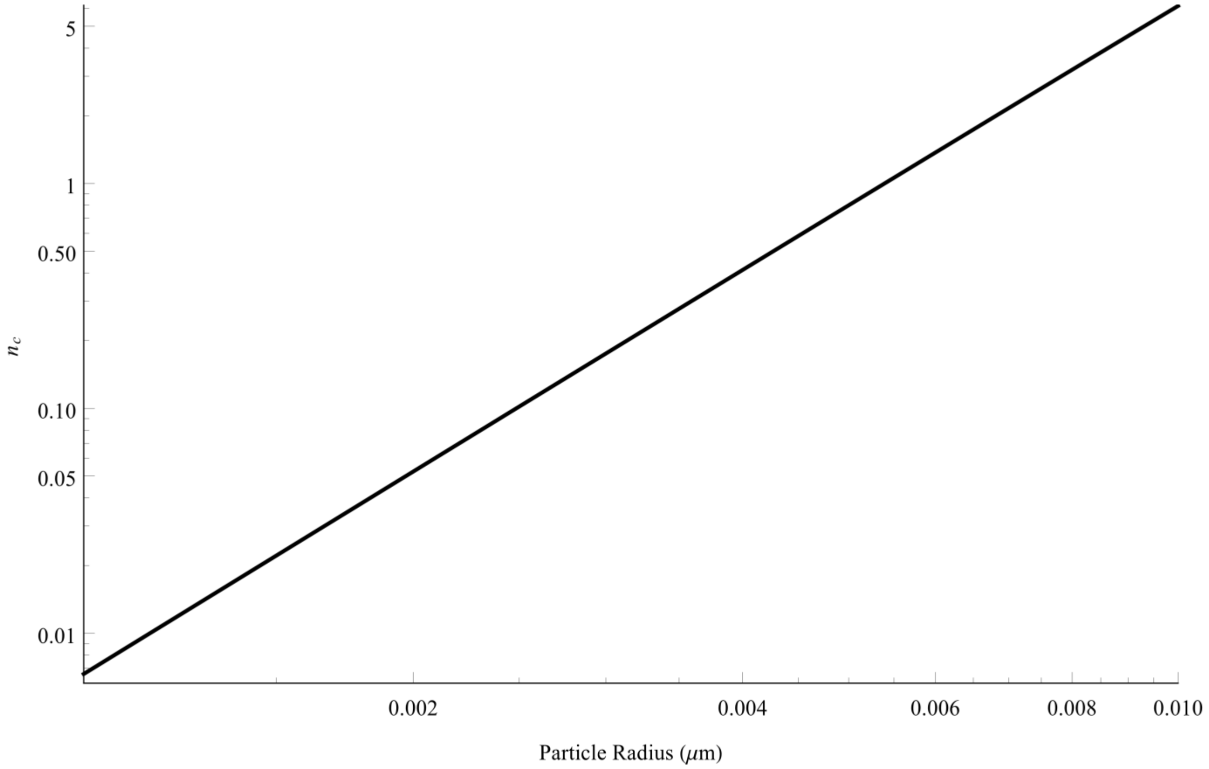


Figure 2 Number of clusters as a function of particle radii in the free molecular regime. See table 1 for the geometry and time step.

III. Clustering Algorithm

While there are a variety of clustering algorithms available, we have focused our attention on the PAM algorithm due to its wide use and computational speed. This algorithm partitions the particles into n_c clusters by minimizing the overall dissimilarity between the n_c representative particles (called medoids) and the other particles. For a detailed description of this algorithm, the reader is referred to the literature[16,18] #, however, a brief explanation of this process is described here.

The program input is the number of clusters (n_c), the data set, and the desired distance metric (in our case, Euclidean distance). The algorithm consists of two phases: a build phase and a swap phase. In the build phase, n_c random particles are chosen to be the initial medoids. A distance matrix is used to associate each other particle to its nearest medoid. In this way, an initial clustering is created with each particle associated with its nearest medoid.

Once the initial cluster is formed, the program then enters the swap phase. Here, the algorithm seeks to improve the existing cluster by determining whether another arrangement of medoids will decrease the total "error"- that is, the distance of each point to its medoid. To this end, for each cluster, the average distance $\overline{\mathcal{D}}_m$ of each particle to its medoid m is determined. Then, each particle i in the cluster is in turn swapped out as the acting medoid and the average distance ($\overline{\mathcal{D}}_i$) is calculated. If another particle gives a lesser mean distance than the original medoid (i.e. $\overline{\mathcal{D}}_i < \overline{\mathcal{D}}_m$), this particle replaces m as the medoid for that cluster. Subsequently, if a medoid replacement occurred in any of the clusters, the particles are re-clustered with regard to the new medoids and the swap phase is repeated. Otherwise, the algorithm is terminated and the completed cluster is returned.

There are a number of alternative implementations that may be used in the above process. The number of iterations within the program (number of times the swap phase is repeated) can be made a fixed value. While this may restrict the quality of the clusters, it can substantially speed the clustering process. Another option is to feed the program an initial set of medoids instead of allowing it to create a random selection. By providing a well distributed representative set of points for the medoids, the user may be able to improve the cluster and the speed with which it is created.

IV. Diffusion Step

Within a given time step, Δt, the particles are moved by Brownian diffusion. First, the diffusion coefficient for each particle is calculated according to Eq (1). Once this is determined for each particle p, its Cartesian coordinates within the sphere are incremented. This is accomplished by performing a random sampling along the normal distributions

$$N\left(x_{old}, \sqrt{2D_p \cdot \Delta t_{Brown}}\right) \tag{5}$$

$$N\left(y_{old}, \sqrt{2D_p \cdot \Delta t_{Brown}}\right) \tag{6}$$

$$N\left(z_{old}, \sqrt{2D_p \cdot \Delta t_{Brown}}\right) \tag{7}$$

to acquire each particle’s new coordinate position which must then be evaluated for possible deposition. For this work, if the new particle location lies on or is outside the geometry, the particle is assumed to have deposited and is removed from the DSMC particle sample list.

This algorithm was verified by comparing it to a finite differenced solution (described in section V) to the equation

$$\frac{\partial Q_\ell}{\partial t} = \nabla^2 Q_\ell(\vec{\boldsymbol{r}}, t) \cdot D_\ell \tag{8}$$

in the case of a sphere of radius $R = .001m$. The other parameters of the simulation follow that of the benchmark described in section VI. The number distribution of the particles at times $t = 10, 20s$ is compared in figure 3 and 4 with the number of particles deposited in time given in figure 5.

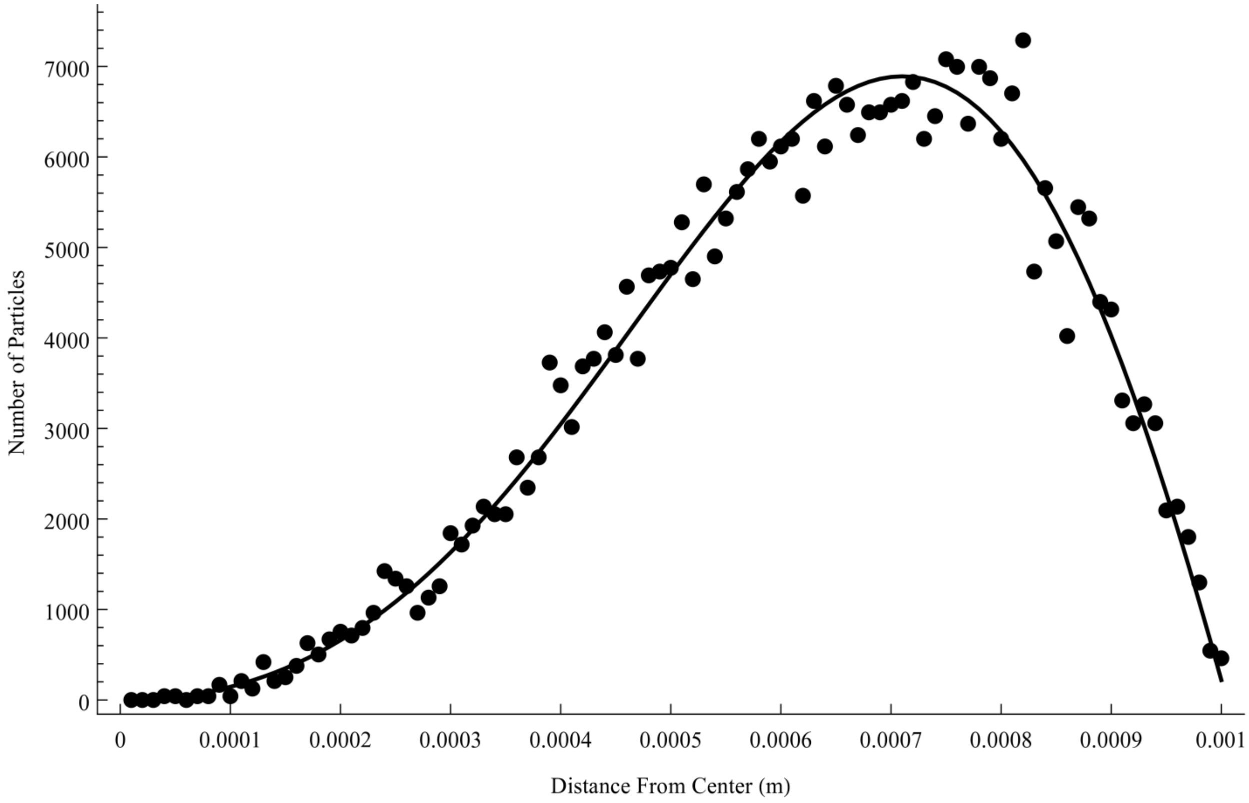


Figure 3 The number of particles a given distance from the center of a R=.001m *sphere at time* t=10s*.*

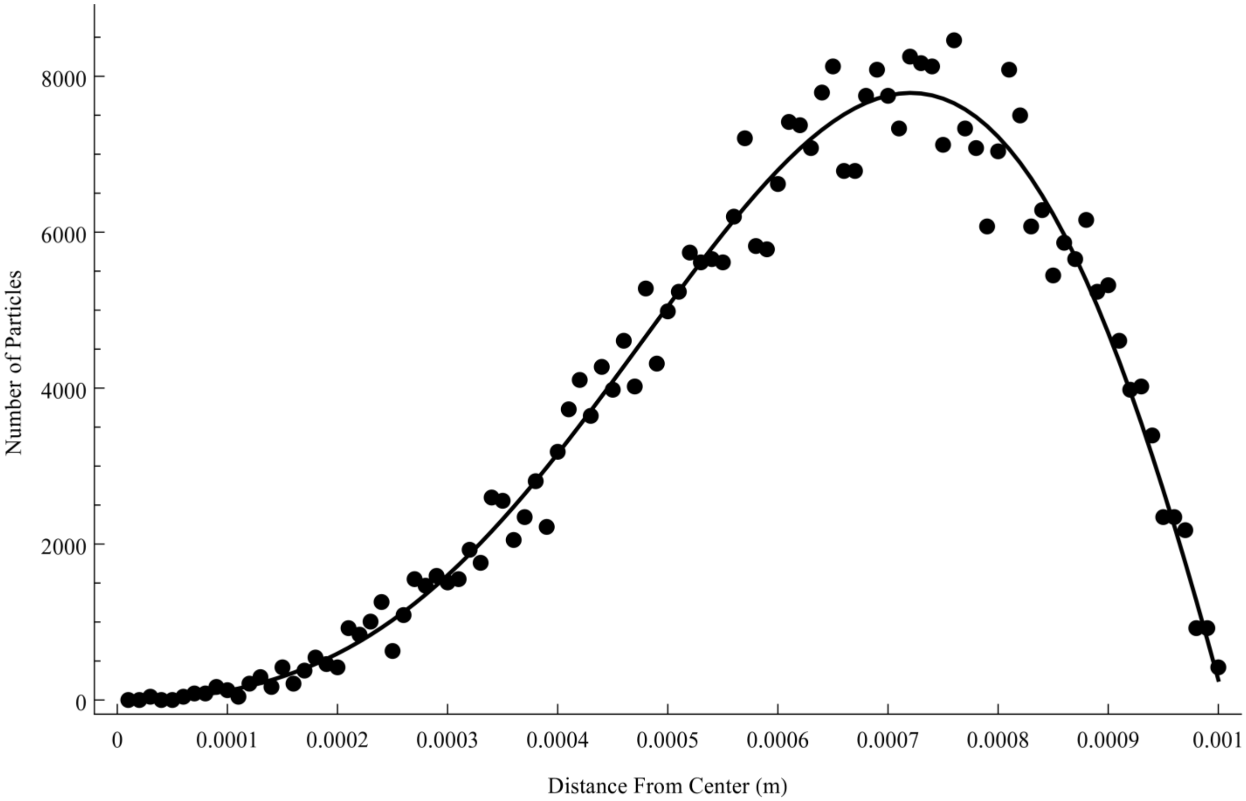


Figure 4 The number of particles a given distance from the center of a R=.001m sphere at time t=20s.

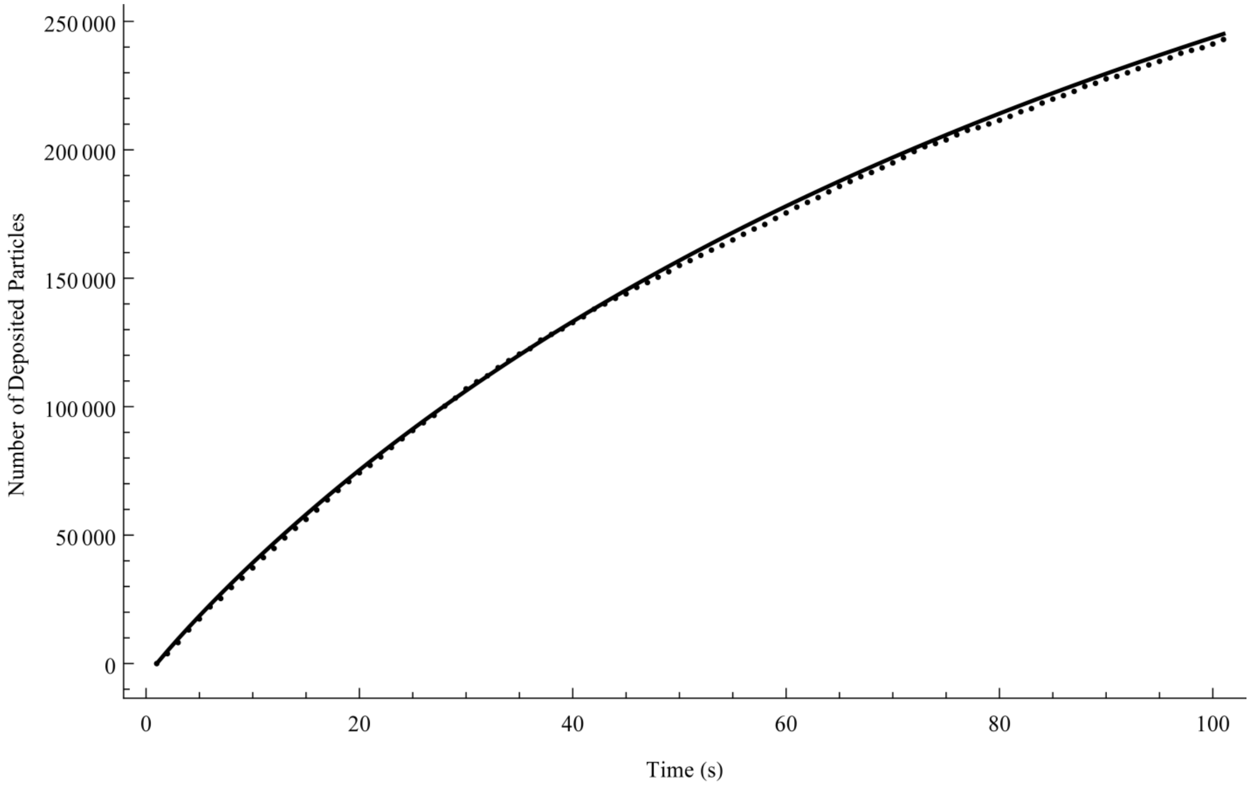


Figure 5 The number of particles deposited on the inner surface of the sphere at a given time.

V. Sectional Method

To verify the DSMC simulations, we have compared the results obtained against those obtained by the sectional technique [ ]. We consider a single component aerosol in spherical geometry and use a volumetric cell balancing technique. The sectional method is employed by MAEROS [ ] in the MELCOR [ ] and CONTAIN [ ] codes. This method solves for the growth of a homogeneous aerosol by converting the general dynamic equation for coagulation[20]

(#)

$$\begin{aligned}\frac{\partial}{\partial t} n(\boldsymbol{v},\boldsymbol{m},t) = \frac{1}{2}\int_0^\infty d\boldsymbol{v}\int_0^\infty d\boldsymbol{w}\int_0^\infty d\boldsymbol{q}\int_0^\infty d\boldsymbol{s} \times n(\boldsymbol{u},\boldsymbol{q},t)\, n(\boldsymbol{w},\boldsymbol{s},t)\, K(u,q|w,s) \\ \times \prod_{p=1}^{N} \delta\big(v_p-u_p-w_p\big)\,\delta\big(m_p-q_p-s_p\big) \qquad (9) \\ - n(\boldsymbol{v},\boldsymbol{m},t)\int_0^\infty d\boldsymbol{u}\int_0^\infty d\boldsymbol{q}\, K(u,q|v,m)\, n(\boldsymbol{u},\boldsymbol{q},t)\end{aligned}$$

into a system of ordinary differential equations by dividing the range of particle diameters into m sections and making assumptions concerning the nature of the number and component distributions [ ]. For a single component aerosol, the assumption made concerning the number distribution is that it can be related to is the total mass within section ℓ, Q_ℓ by the equation $n(\vec{r},v,t) = \frac{Q_\ell(\vec{r},t)f'(v)}{v(x_\ell - x_{\ell-1})}$ for some "conveniently" chosen size variable $x = f(v)$ for which the distribution is constant over each section. Given this, Eq. (9) becomes

$$\begin{aligned}\left(\frac{dQ_\ell}{dt}\right)_{coag} = \frac{1}{2}\sum_{i=1}^{\ell-1}\sum_{j=1}^{\ell-1} Q_i Q_j [\bar{\beta}^{1a}_{i,j,\ell} + \bar{\beta}^{1b}_{i,j,\ell}] - Q_\ell \sum_{i=1}^{\ell-1} Q_i\big[\bar{\beta}^{2a}_{i,\ell} - \bar{\beta}^{2b}_{i,\ell}\big] - \frac{1}{2}\bar{\beta}^{3}_{\ell,\ell}(Q_\ell)^2 \qquad (10) \\ - Q_\ell \sum_{i=\ell+1}^{m} \bar{\beta}^{4}_{i,\ell} Q_i.\end{aligned}$$

where

$$
\begin{aligned}
\bar{\beta}^{1a}_{i,j,\ell} &= \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\theta(v_{\ell-1} < u+v < v_\ell)\, u\, \beta(u,v)}{u\, v\, (x_i - x_{i-1})\big(x_j - x_{j-1}\big)} dydx \\
\bar{\beta}^{1b}_{i,j,\ell} &= \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\theta(v_{\ell-1} < u+v < v_\ell)\, v\, \beta(u,v)}{u\, v\, (x_i - x_{i-1})\big(x_j - x_{j-1}\big)} dydx \\
\bar{\beta}^{2a}_{i,\ell} &= \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\theta(u+v > v_\ell)\, u\, \beta(u,v)}{u\, v\, (x_i - x_{i-1})(x_\ell - x_{\ell-1})} dydx \\
\bar{\beta}^{2b}_{i,\ell} &= \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\theta(u+v < v_\ell)\, v\, \beta(u,v)}{u\, v\, (x_i - x_{i-1})(x_\ell - x_{\ell-1})} dydx \\
\bar{\beta}^{3}_{\ell,\ell} &= \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{\theta(u+v > v_\ell)\, (u+v)\, \beta(u,v)}{u\, v\, (x_\ell - x_{\ell-1})(x_\ell - x_{\ell-1})} dydx \\
\bar{\beta}^{4}_{i,\ell} &= \int_{x_{i-1}}^{x_i}\int_{x_{j-1}}^{x_j} \frac{u\, \beta(u,v)}{u\, v\, (x_i - x_{i-1})(x_\ell - x_{\ell-1})} dydx.
\end{aligned}
\tag{11}
$$

And the “betas” are the coagulation kernels, and “theta” is the Heaviside function. In the case of a heterogeneous aerosol (we are assuming that the spatial environment is uniform and the spatial heterogeneity is only with respect the aerosol particle distribution), the sectional equation of interest is

$$
\frac{\partial Q_\ell(\vec{\boldsymbol{r}},t)}{\partial t} = \nabla^2 Q_\ell(\vec{\boldsymbol{r}},t)\cdot D_\ell + \left(\frac{dQ_\ell}{dt}\right)_{coag} \tag{12}
$$

where

$$
Q_\ell(\vec{r},t) = \int_{v_{\ell-1}}^{v_\ell} v\cdot n(\vec{r},v,t)dv. \tag{13}
$$

and

$$D_\ell = \int_{v_{\ell-1}}^{v_\ell} \frac{D\, f'(v)}{(x_\ell - x_{\ell-1})}\, dv. \tag{14}$$

By using a volumetric cell balancing and finite difference approximation, we can construct a system of $I \times m$ equations (equation 16**) which can be solved using a numeric differential equation solver where I is the number of cells we break the sphere into; and V_i and $A_{i+\frac{1}{2}}$ represent the volume of the i^{th} cell and the surface area of the $i+\frac{1}{2}$ shell respectively (see figure 6).

$$\begin{aligned} V_i \frac{\partial Q_\ell(i,t)}{\partial t} &= D_\ell \left[A_{i+\frac{1}{2}} \frac{Q_\ell(i+1,t) - Q_\ell(i,t)}{h_r} - A_{i-\frac{1}{2}} \frac{Q_\ell(i,t) - Q_\ell(i-1,t)}{h_r} \right] \\ &\quad + V_i \left(\frac{\partial Q_\ell(i,t)}{\partial t} \right)_{coag}. \end{aligned} \tag{15}$$

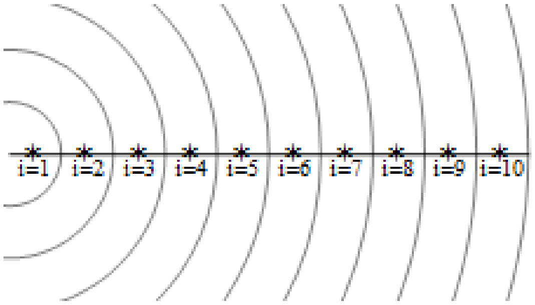


Figure 6 A sphere of radius R broken into 10 cells.

VI. Benchmark Simulation

To benchmark the DSMC clustering program it was compared to the aforementioned cell balance sectional method for a simulation over a sphere. We assume a one centimeter diameter sphere is filled with air at STP and an initial distribution of aerosol particles is evenly dispersed (with respect to number per unit volume) within the sphere at time $t = 0$ with the initial particle size distribution taken to be a log-normal distribution

$$p(d) = \frac{1}{\sqrt{2\pi\sigma}}\frac{1}{d}Exp\left[-\frac{1}{2\sigma}\left(Log\left[d/\bar{d}\ \right]\right)^2\right] \tag{16}$$

for $\bar{\mathrm{d}} = 0.1$ μm and $\sigma = 0.035$ (see figure 7**). Particles are assumed to diffuse solely by means of Brownian motion and those coming in contact with the inner sphere surface deposit. We have compared the rate of particle deposition as well as the cumulative particle size distribution in time for the two simulations. For DSMC, a sample list of length 10^5 is generated by applying a selection-rejection technique over the size distribution and assigning each particle a random position within the sphere. The results of eight distinct Monte Carlo simulations are averaged to provide the uncertainty in the results.

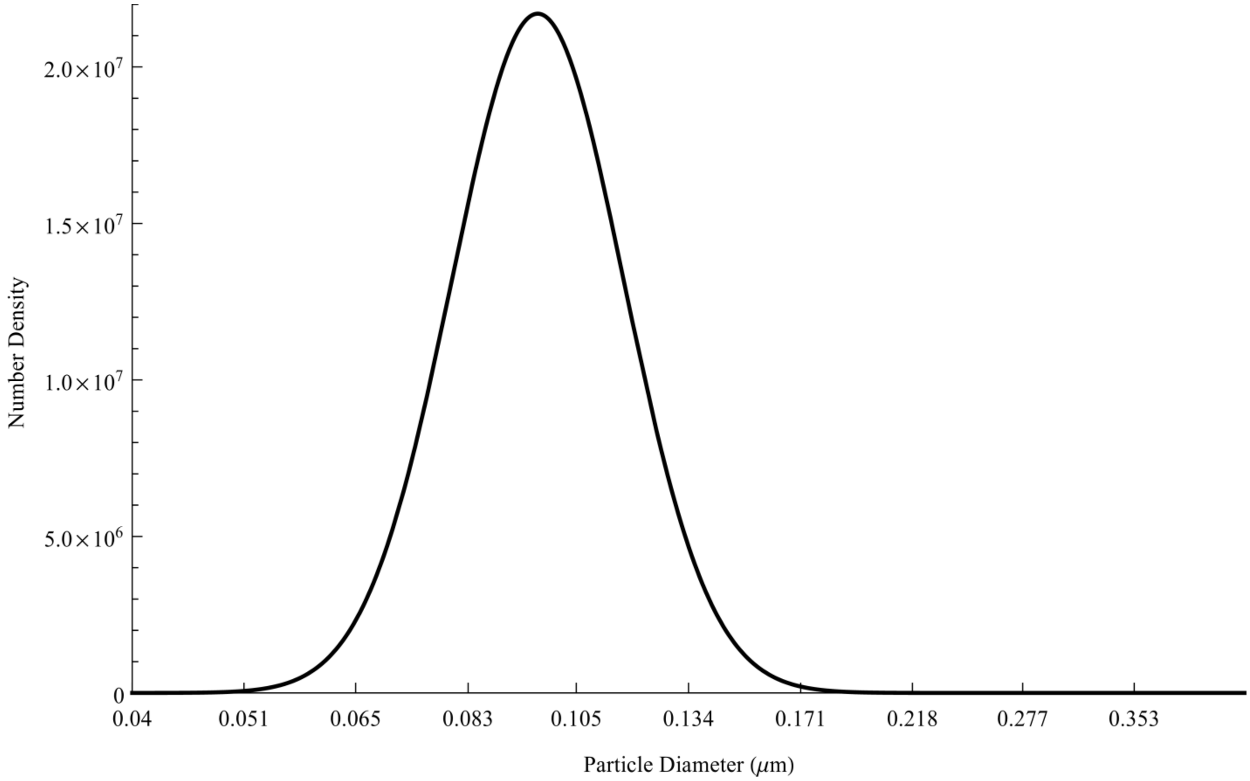


Figure 7 The initial size distribution for the benchmark simulation. The curve is the log-normal distribution $P(d) = \frac{1}{\sqrt{2\pi\sigma}}\frac{1}{d}Exp\left[\frac{-1}{2\sigma}\left(Log\left[d/\bar{d}\;\right]\right)^2\right]$ *for* $\bar{d} = 0.1$ *μm,* $\sigma = 0.035$*. This curve is used to generate the initial conditions for the cell balance and DSMC simulations.*

In the first few seconds of the simulation, there is a sharp jump in particle deposition. The cell balance technique requires a large number of cells to properly model the deposition rate onto the sphere surface. In figure eight,** a plot of the mass deposited in time demonstrates the convergence of the cell balance simulation for increasing values of I. Based upon this figure, the number of cells chosen for the benchmark simulation was taken to be $I = 100$. In tables 1-3, the initial parameters for the benchmark simulation are provided.

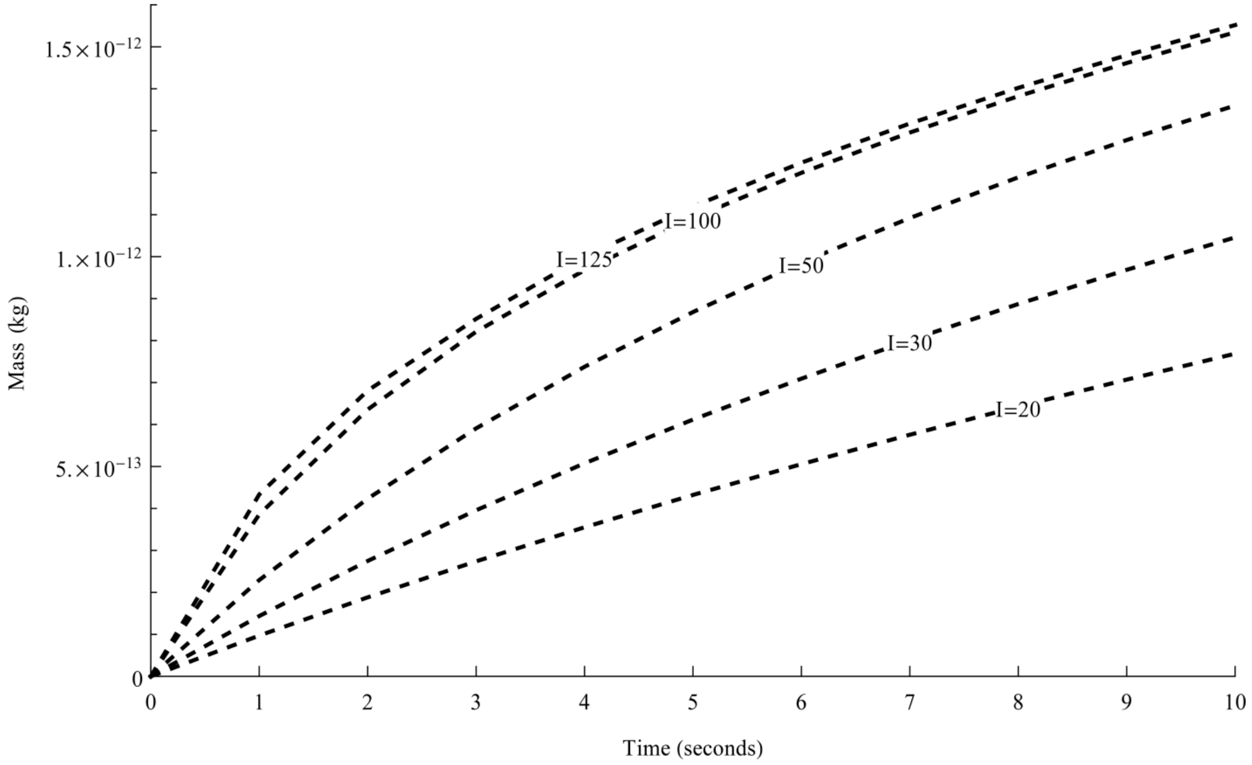


Figure 8 The mass in kilograms deposited onto the surface of the sphere for increasing values of I*. We note that the valuse converge as* I *increases. This figure serves as justification of our choice of* $I = 100$ *cells in the benchmark simulation.*

Table 1 General initial parameters for the benchmark simulation.

General Parameters		
Sphere Radius	$R = 0.5$	cm
Particle Number Density	$N_p = 1 \times 10\text{^}14$	$particles/m\text{^}3$
Particle Mass Density	$\rho = 1000$	$kg/m\text{^}3$
Number of Particles	$n = 5.24 \times 10\text{^}7$	
End Time	$T = 20$	sec

Table 2 Initial parameters for the cell balanced sectional simulation of the benchmark problem.

Cell Parameters		
Minimum Size	$\ell min = 0.04$	μm
Maximum Size	$\ell max = 0.45$	μm
Number of Divisions	$m = 20$	
Number of Components	$ktotal = 1$	
Number of Cells	$I = 100$	
Number Distribution	$f(v) = Log(d(v))$	

	Where d(v) is the diameter of the particle in microns.

Table 3 Initial parameters for the DSMC cluster simulation of the benchmark problem.

DSMC Parameters	
Number of Initial Simulated Particles	$N = 1.0 \times 10\text{^}5$
Scale Factor	523.6
Time Steps	$\Delta t = 0.1$ sec
Initial Number of Clusters	$n_C = 223^*$
*While this is the initial number of clusters, within the simulation, the value is allowed to adapt in time to be the minimum of equations 3 and 4**.	

The effectiveness of the clustering algorithm can be demonstrated by observing the rate at which coagulation occurs and comparing it with the sectional simulation. This can be inferred by comparing the size distributions of the suspended particles at various times. In figures 9** and 10, the suspended mass in each of the 20 size sections is compared at times $t = 10, 20\ s$. We note that the distributions agree well at both times implying that the cluster algorithms can successfully be used in conjunction with DSMC to model aerosol coagulation. Tables 4 and 5 give the results of the two programs and the uncertainty corresponding to the 8 DSMC simulations.

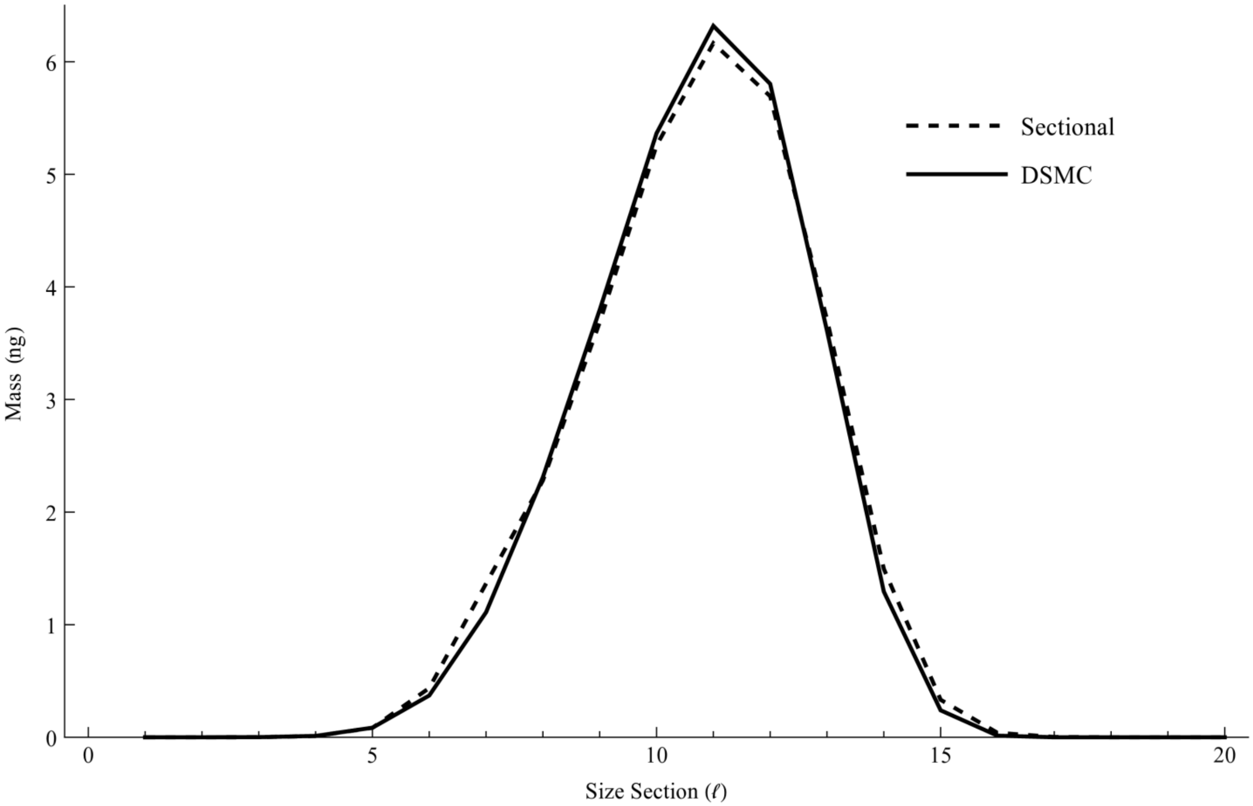


Figure 9 The mass in ng suspended at time t=10*s in each of the* 20 *size sections for DSMC and sectional simulations.*

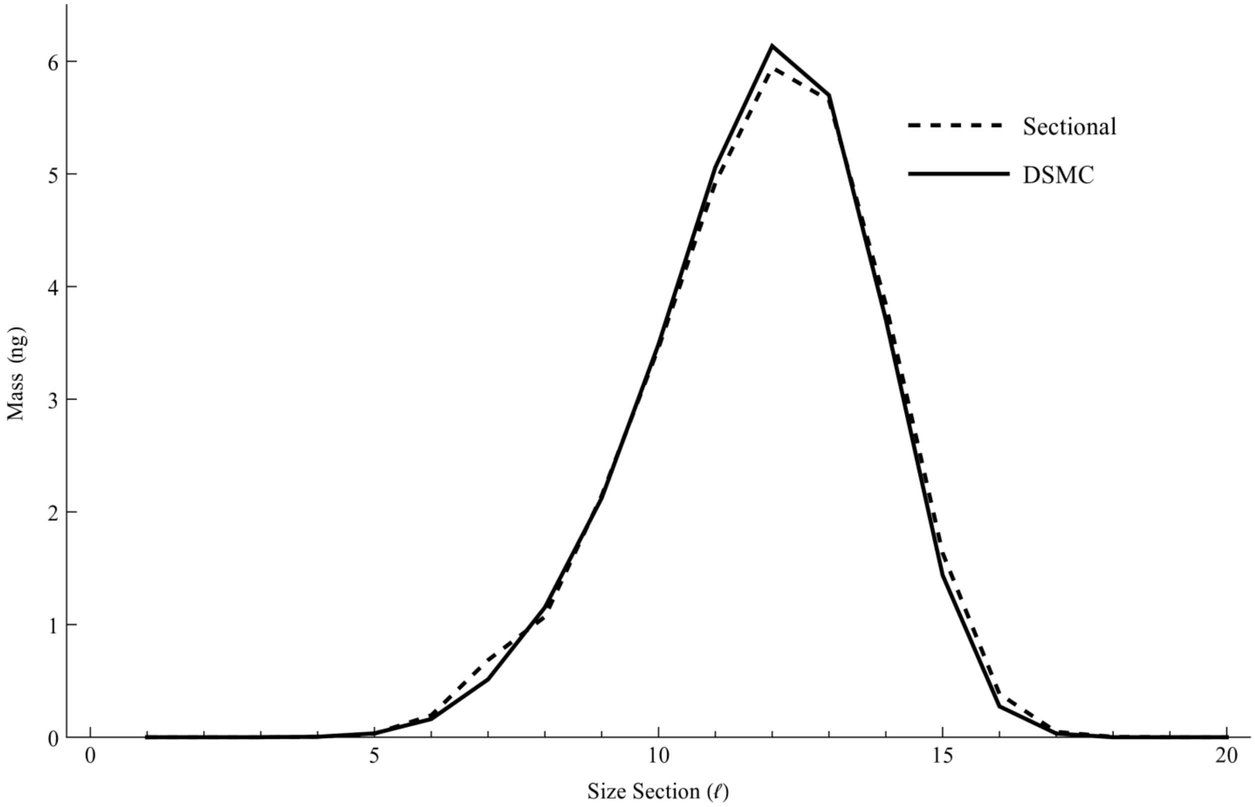
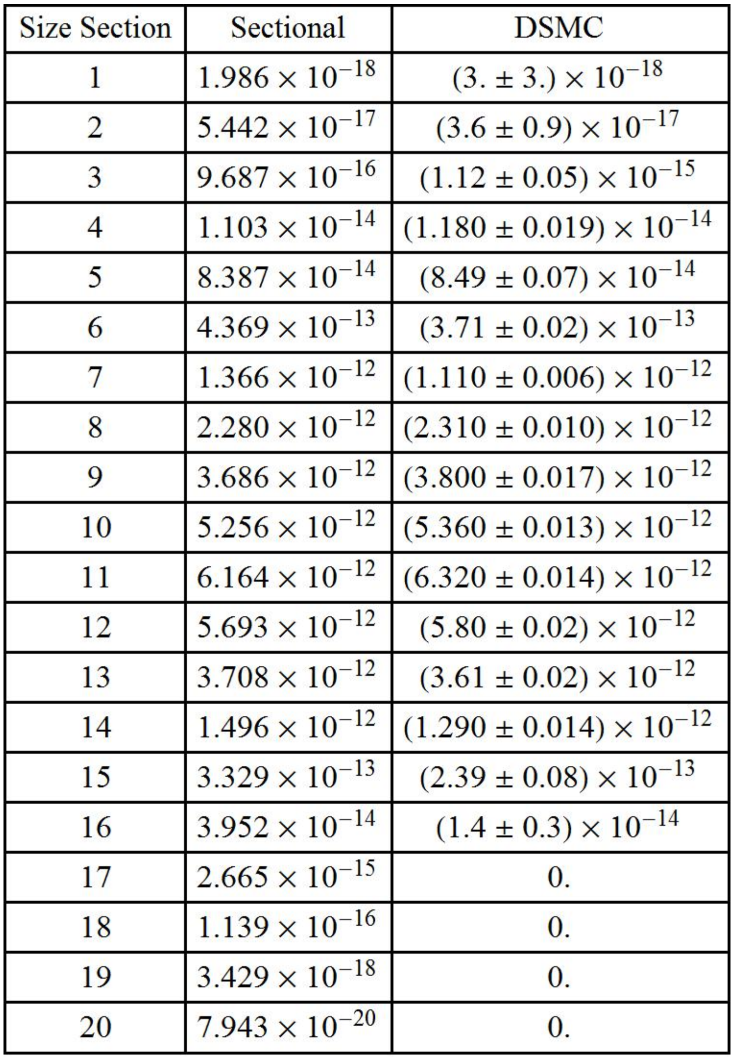


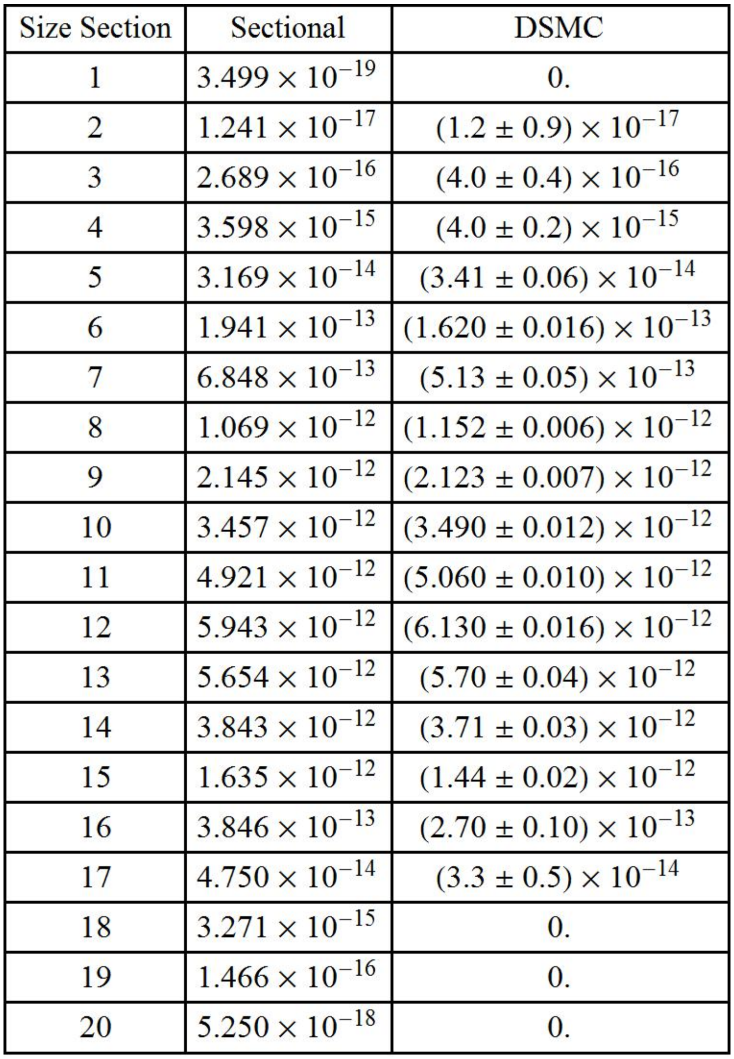
Figure 10 The mass in ng suspended at time t=20s in each of the 20 size sections for DSMC and sectional simulations.

Table 4 The total mass in kg within the each size section suspended in the sphere at time t=10*s.*



Size Section	Sectional	DSMC
1	1.986×10^{-18}	$(3. \pm 3.) \times 10^{-18}$
2	5.442×10^{-17}	$(3.6 \pm 0.9) \times 10^{-17}$
3	9.687×10^{-16}	$(1.12 \pm 0.05) \times 10^{-15}$
4	1.103×10^{-14}	$(1.180 \pm 0.019) \times 10^{-14}$
5	8.387×10^{-14}	$(8.49 \pm 0.07) \times 10^{-14}$
6	4.369×10^{-13}	$(3.71 \pm 0.02) \times 10^{-13}$
7	1.366×10^{-12}	$(1.110 \pm 0.006) \times 10^{-12}$
8	2.280×10^{-12}	$(2.310 \pm 0.010) \times 10^{-12}$
9	3.686×10^{-12}	$(3.800 \pm 0.017) \times 10^{-12}$
10	5.256×10^{-12}	$(5.360 \pm 0.013) \times 10^{-12}$
11	6.164×10^{-12}	$(6.320 \pm 0.014) \times 10^{-12}$
12	5.693×10^{-12}	$(5.80 \pm 0.02) \times 10^{-12}$
13	3.708×10^{-12}	$(3.61 \pm 0.02) \times 10^{-12}$
14	1.496×10^{-12}	$(1.290 \pm 0.014) \times 10^{-12}$
15	3.329×10^{-13}	$(2.39 \pm 0.08) \times 10^{-13}$
16	3.952×10^{-14}	$(1.4 \pm 0.3) \times 10^{-14}$
17	2.665×10^{-15}	0.
18	1.139×10^{-16}	0.
19	3.429×10^{-18}	0.
20	7.943×10^{-20}	0.

Table 5 The total mass in kg within the each size section suspended in the sphere at time t=20s.



Size Section	Sectional	DSMC
1	3.499×10^{-19}	0.
2	1.241×10^{-17}	$(1.2 \pm 0.9) \times 10^{-17}$
3	2.689×10^{-16}	$(4.0 \pm 0.4) \times 10^{-16}$
4	3.598×10^{-15}	$(4.0 \pm 0.2) \times 10^{-15}$
5	3.169×10^{-14}	$(3.41 \pm 0.06) \times 10^{-14}$
6	1.941×10^{-13}	$(1.620 \pm 0.016) \times 10^{-13}$
7	6.848×10^{-13}	$(5.13 \pm 0.05) \times 10^{-13}$
8	1.069×10^{-12}	$(1.152 \pm 0.006) \times 10^{-12}$
9	2.145×10^{-12}	$(2.123 \pm 0.007) \times 10^{-12}$
10	3.457×10^{-12}	$(3.490 \pm 0.012) \times 10^{-12}$
11	4.921×10^{-12}	$(5.060 \pm 0.010) \times 10^{-12}$
12	5.943×10^{-12}	$(6.130 \pm 0.016) \times 10^{-12}$
13	5.654×10^{-12}	$(5.70 \pm 0.04) \times 10^{-12}$
14	3.843×10^{-12}	$(3.71 \pm 0.03) \times 10^{-12}$
15	1.635×10^{-12}	$(1.44 \pm 0.02) \times 10^{-12}$
16	3.846×10^{-13}	$(2.70 \pm 0.10) \times 10^{-13}$
17	4.750×10^{-14}	$(3.3 \pm 0.5) \times 10^{-14}$
18	3.271×10^{-15}	0.
19	1.466×10^{-16}	0.
20	5.250×10^{-18}	0.

While figures 9 and 10** reveal that the clustering algorithm is able to accurately model coagulation, it must also be shown that it does not interfere with the modeling of the aerosol motion. In figure 11**, the total mass deposited on the surface of the sphere is plotted versus time in seconds. In the initial few seconds of the simulation, DSMC shows a faster rate of deposition than sectional. As discussed earlier in regards to figure 8**, this is due at least in part to the choice in the number of cells, I. A greater number of cells would reduce this difference, however it would be computationally limiting. As time continues, the sectional simulation overtakes DSMC in the total amount of deposited mass. This is most likely due to the statistical error from the depletion of the particle list. As table 6 demonstrates, however, the sectional results remain within the bounds of the uncertainty of DSMC.

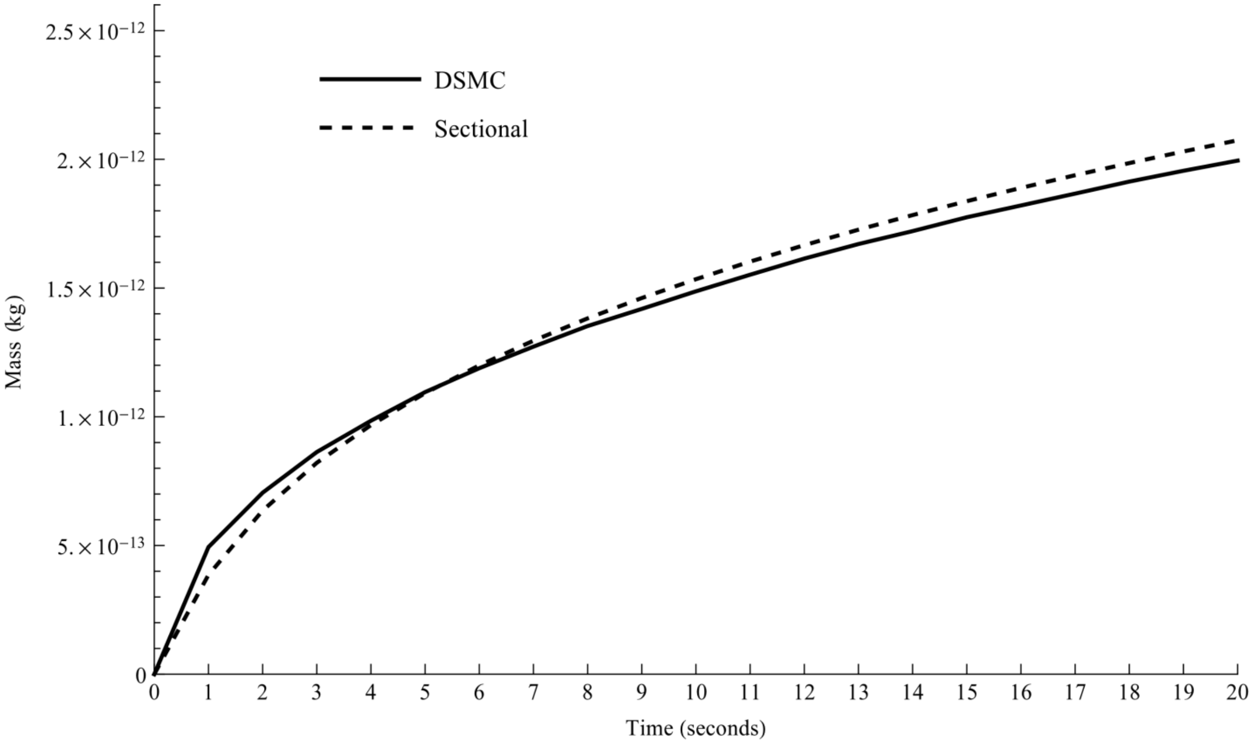
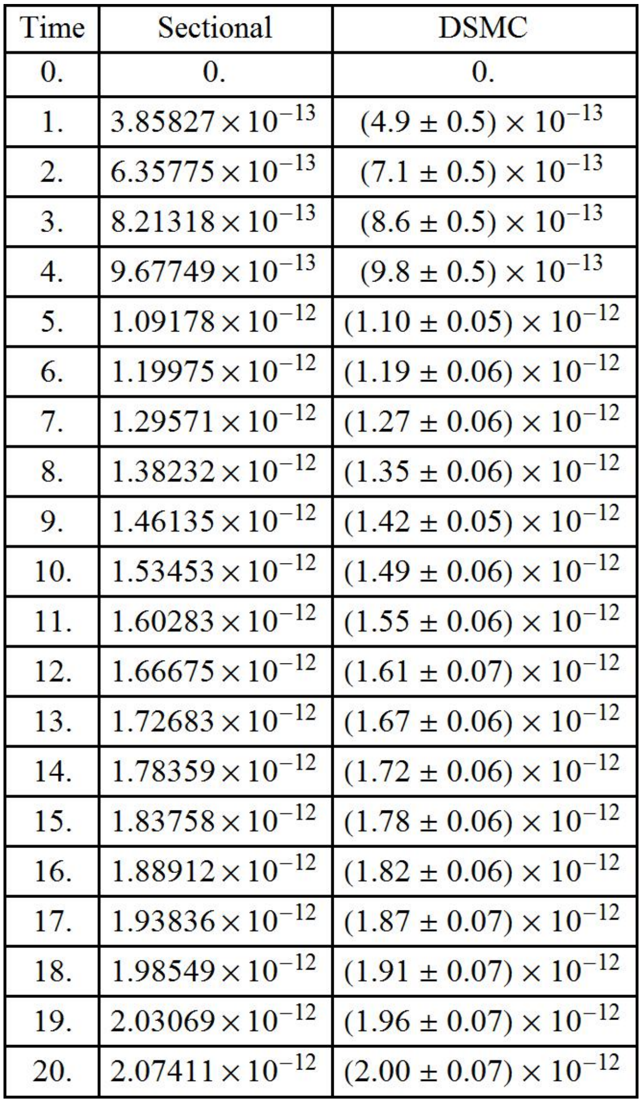


Figure 11 The total mass in kg *deposited on the inner surface of the sphere for the DSMC and sectional simulations in time.*

Table 6 The total mass in kilograms deposited onto the inner surface of the sphere at a given time (s).



Time	Sectional	DSMC
0.	0.	0.
1.	3.85827×10^{-13}	$(4.9 \pm 0.5) \times 10^{-13}$
2.	6.35775×10^{-13}	$(7.1 \pm 0.5) \times 10^{-13}$
3.	8.21318×10^{-13}	$(8.6 \pm 0.5) \times 10^{-13}$
4.	9.67749×10^{-13}	$(9.8 \pm 0.5) \times 10^{-13}$
5.	1.09178×10^{-12}	$(1.10 \pm 0.05) \times 10^{-12}$
6.	1.19975×10^{-12}	$(1.19 \pm 0.06) \times 10^{-12}$
7.	1.29571×10^{-12}	$(1.27 \pm 0.06) \times 10^{-12}$
8.	1.38232×10^{-12}	$(1.35 \pm 0.06) \times 10^{-12}$
9.	1.46135×10^{-12}	$(1.42 \pm 0.05) \times 10^{-12}$
10.	1.53453×10^{-12}	$(1.49 \pm 0.06) \times 10^{-12}$
11.	1.60283×10^{-12}	$(1.55 \pm 0.06) \times 10^{-12}$
12.	1.66675×10^{-12}	$(1.61 \pm 0.07) \times 10^{-12}$
13.	1.72683×10^{-12}	$(1.67 \pm 0.06) \times 10^{-12}$
14.	1.78359×10^{-12}	$(1.72 \pm 0.06) \times 10^{-12}$
15.	1.83758×10^{-12}	$(1.78 \pm 0.06) \times 10^{-12}$
16.	1.88912×10^{-12}	$(1.82 \pm 0.06) \times 10^{-12}$
17.	1.93836×10^{-12}	$(1.87 \pm 0.07) \times 10^{-12}$
18.	1.98549×10^{-12}	$(1.91 \pm 0.07) \times 10^{-12}$
19.	2.03069×10^{-12}	$(1.96 \pm 0.07) \times 10^{-12}$
20.	2.07411×10^{-12}	$(2.00 \pm 0.07) \times 10^{-12}$

VII. Conclusions

An alternative method for modeling a heterogeneous aerosol was proposed using a Partitioning Around the Medoids clustering algorithm instead of a traditional mesh for a Monte Carlo simulation. A benchmark simulation was then constructed for demonstrating the fidelity of this mesh-free approach as compared to a cell balanced sectional technique. It was shown that the method gave similar results to its analogous sectional simulation lending confidence to the accuracy of the method. The true value of the mesh-free approach will be in complex geometries where the quality of a mesh is limited by its necessary complexity.

VIII. Acknowledgments

This work has been supported by a grant NEUP-965, by the U.S. Department of Energy. Shawn Campbell was also supported by a GAANN grant from the U.S. Department of Education and a grant from the U.S. Nuclear Regulatory Commission.

References:

1. K. K. MURATA D. C. WILLIAMS, J. TILLS, R. O. GRIFFITH, R. G. GIDO, E. L. TADIOS, F. J. DAVIS, G. M. MARTINEZ, K. E. WASHINGTON, “Code Manual for CONTAIN 2.0: A Computer Code for Nuclear Reactor Containment Analysis,” Sandia National Laboratories, Albuquerque, NM, Prepared for the U.S. Nuclear Regulatory Commission, NUREG/CR-6533, SAND97-1735 (1997)
2. R. O. GAUNTT, R. K. COLE, C. M. ERICKSON, R. G. GIDO, R. D. GASSER, S. B. RODRIGUEZ, M. F. YOUNG, “MELCOR Computer Code Manuals,” Sandia National Laboratories, Albuquerque, NM, Prepared for the U.S. Nuclear Regulatory Commission, NUREG/CR-6119, SAND2000-2417/1 (2000)
3. M. M. R. WILLIAMS and S. K. LOYALKA, “Aerosol Science: Theory and Practice with Special Applications to the Nuclear Industry,” Pergamon, Oxford (1991).
4. R. SHER and R. R. HOBBINS, “Transport and Removal of Aerosols in Nuclear Power Plants Following Severe Accidents,” American Nuclear Society (2011)
5. W. MOE, "HTGR Mechanistic Source Terms White Paper," Idaho National Laboratory (United States). Funding organization: DOE-NE (United States), INL/EXT-10-17997 (2010).
6. IAEA, “Fuel Performance and Fission Product Behaviour in Gas Cooled Reactors,” IAEA-TECDOC-978 (1997)
7. R. N. MORRIS, "Next Generation Nuclear Plant Phenomena Identification and Ranking Tables (PIRTs) Volume 3: Fission-Product Transport and Dose PIRTs," Oak Ridge National Laboratory (ORNL), (2008)

8. P. W. HUMRICKHOUSE, "HTGR Dust Safety Issues and Needs for Research and Development," INL/EXT-11-21097, Idaho National Laboratory, Idaho Falls, ID2011

9. S. K. FRIEDLANDER, “Smoke, Dust, and Haze: Fundamentals of Aerosol Dynamics,” 2nd ed., New York: Oxford University Press, (2000).

10. W. HINDS, “Aerosol Technology: Properties, Behavior, and Measurement of Airborne Particles,” John Wiley & Sons Inc. (1982)

11. N. A. FUCHS, “The Mechanics of Aerosols,” Revised ed., Pergamon Press New York (1964)

12. E. J. DAVIS and G. SCHWEIGER, “The Airborne Microparticle: its Physics, Chemistry, Optics, and Transport Phenomena,” Springer, (2002)

13. F. GELBARD, and J. H. SEINFELD, “Simulation of Multicomponent Aerosol Dynamics,” *J. of Colloid. and Interface Sci.*, **78**(2), 485-501 (1980)

14. F. GELBARD, “Modeling Multicomponent Aerosol Particle Growth by Vapor Condensation,” *Aerosol Sci. and Tech.*, **12**(2), 399-412 (1990)

15. F. GELBARD, “MAEROS User Manual”, Sandia National Laboratories, NUREG/CR-1391, SAND 80-0822 (1982)

16. S. K. LOYALKA “Direct Simulation of Multi-Component Aerosol Dynamics,” *Trans. Am. Nucl. Soc.,* **88**, 334 (2003)

17. G. PALANISWAAMY and S. K. LOYALKA, “Direct Simulation Monte Carlo Multi-Component Aerosol Dynamics: Coagulation, Deposition and Source Reinforcement,” *Nucl. Tech.,* **160**(2), 187-204 (2007)

18. G. PALANISWAAMY and S. K. LOYALKA, “Direct Simulation Monte Carlo Aerosol Dynamics: Collisional Sampling Algorithms,” *Ann. Nucl. Energy* **34**(1), 13-21 (2007)

19. G. PALANISWAAMY and S. K. LOYALKA, “Direct Simulation Monte Carlo Aerosol Dynamics: Coagulation and Condensation,” *Ann. Nucl. Energy*, **35**(3), 485-494 (2008)

20. J. PALSMEIER and S.K. LOYALKA, “Evolution of Charged Aerosols: Role of Charge on Coagulation,” *Nucl. Tech.*, **184**(1), 78-95 (2013)

21. M. P. SIMONES and S. K. LOYALKA, “Measurements of Charged Aerosol Coagulation,” *Nuc. Tech.*, **189**, 45-62 (2015)

22. J.W. PARK and S.K. LOYALKA, “Role of Spatial Inhomogeneities in Source Term Aerosol Dynamics”, *Nuc. Sci and Eng.*, **101**, 269-279 (1989)

23. Y. EFENDIEV and M. R. ZACHARIAH, “Hybrid Monte Carlo Method for Simulation of Two-Component Aerosol Coagulation and Phase Segregation,” *J. of Colloid. and Interface Sci.*, **249**, 30-43 (2002)

24. Z. SUN, R. L. AXELBAUM, and J. I. HUERTAS, “Monte Carlo Simulation of Multicomponent Aerosols Undergoing Simultaneous Coagulation and Condensation,” *Aerosol Sci. and Tech.*, **38**(10), 963–971 (2004)

25. M. SMITH, K. LEE and T. MATSOUKAS, “Coagulation of charged aerosols,” *J. of Nanoparticle Res.*, **1**, 185–195 (1999).

26. J. WEI, “A Monte Carlo Method for Coagulation of Charged Particles,” *J. of Aerosol Sci.*, **65**, 21-25 (2013)

27. G. A. BIRD, “Molecular Gas Dynamics,” Oxford University Press, London (1976).

28. G. A. BIRD, “Molecular Gas Dynamics and Direct Simulation of Gas Flows,” 2nd ed. Oxford University Press, London (1994)

29. S. CAMPBELL, S. K. LOYALKA, “Computation of Aerosol Evolution under Coagulation,” *Nuc. Sci and Eng.*, Accepted for publication Feb 2015